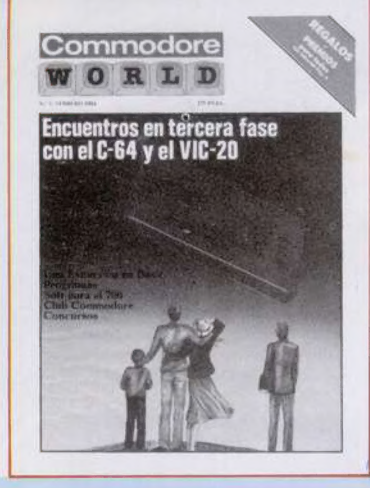
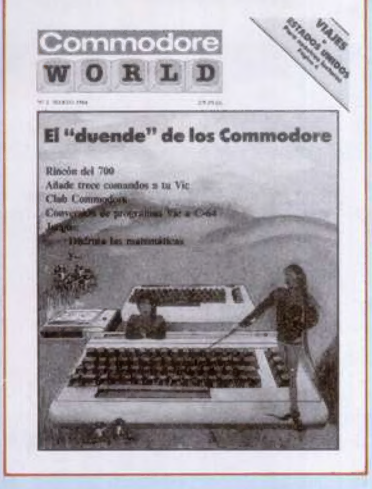
Commodore WORLD

Nº 5. JUNIO 1984 275 PTAS.

Primer Concurso Nacional Escolar de Programación
CONVOCATORIA Y BASES (Pág. 6)











"Commodore World" aparte de venderse generalmente en kioscos, se encuentra asimismo a la venta en los siguientes distribuidores de Commodore y librerías.

ALICANTE

Casa Wagner, c/ Juan Carlos I, 37. Elda.
Tel. (965) 39 03 96.
Ferretería Progreso. c/ General Jordana, 28. Elda.
Tel. (965) 38 11 45.

BADAJOZ

Control Sistemas. Avda. Santa Marina, 25A.
Tel. (924) 25 88 00.

BARCELONA

Computer Service. Avda. Abad Marcet, 325. Tarrasa.
Tel. (93) 788 63 77
Librería Emilia Pérez Radua. c/Mayor, 35. Castellar del Vallés. Tel. (93) 714 89 51.
Librería Michel. Ronda Guinardo, 1. Sardañola.
Mozart. c/ Jaime I, 145, Mollet. Tel. (93) 593 75 01

BURGOS

E.I.S.A. c/ Madrid, 4. Tel. (947) 20 46 24.

CANARIAS

Relax. Rambla de Pulido, 85. Santa Cruz de Tenerife.
Tel. (922) 28 37 05.

CIUDAD REAL

CISA. c/ D. Víctor, 15. Tomelloso.
Tels. (926) 51 00 07 - 51 15 72.
Electrónica Turrillo. c/ Pedrera Baja, 7.
Tel. (926) 22 38 67.

LA CORUÑA

Cetronic, S. L. c/ Palomar, 22 Bajo. Tel. (981) 27 26 54.
Photo Copy. c/ Teresa Herrera, 9. Tel. (981) 21 34 21.
Sanlusa, S. L. San Luis, 46 al 50. Tel. (981) 23 07 49.

GERONA

Digit Informática. c/ Banyoles, 13. Barri de Sant Roc.
Olot. Tel. (972) 26 94 01.

HUELVA

Computerlog, S. A. c/ Tendaleras, 15.
Tel. (955) 25 81 99

MADRID

Chips + Tips - C/Puerto Rico, 21-23
Librería Díaz de Santos. c/ Lagasca, 98.
Librería García Pena. c/ Cavanilles, 52.
Librería Ra-Ma. c/ Chiquinquira, 28. Urb. Columbia.
Microtronic. c/ Quinta del Sol, 39. Las Rozas.
Multiprensa. c/ Castelló, 37.
País de los Microordenadores. Bravo Murillo, 18.
Sonitel Central. c/ Clara del Rey, 24.

MURCIA

Procoinsa. Ronda Norte, 27. Tel. (968) 23 94 49.

PALMA DE MALLORCA

AEF. c/ Balmes, 13. Tel. (971) 27 36 66.

PONFERRADA

MicroBierzo. c/ Carlos I, 2. Tel. (987) 41 74 21.

SALAMANCA

Compusoft. c/ Arco, 1.

SANTANDER

Librería Hernández. San Francisco, 15. Tel. (942) 22 53 30

SANTIAGO DE COMPOSTELA

GestyComputer. Avda. Romero Donallo, 25.
Tel. (981) 59 87 54.

SEVILLA

Papelería Mora. c/ Santa Cruz, 5.

TARRAGONA

Oficomplet. Plaza de la Cinta, 6. Tortosa.
Tel. (977) 44 14 50.

ZARAGOZA

ADA Computer. Centro Independencia.
Paseo Independencia, 24-26. Tel. (976) 29 85 62.

TIRADA DE COMMODORE WORLD
Certificación de 28.000 ejemplares

Según certificado firmado por la auditoría internacional DELOITTE HASKINS SELLS, S. A.
el 5 de marzo de 1984,
la tirada de Commodore World es de 28.000 ejemplares.
Este certificado se encuentra en las oficinas de S.I.M.S.A. para comprobación
de cualquier persona o entidad que lo desee.



SUMARIO 5

Junio 1984

CONTENIDO

PAG.

PROXIMO NUMERO

- Los "strings": "Cadena perpetua... ¿o no?"
- Música maestro
- Comienzo del manual castellano del superexpander
- Las desconocidas posibilidades del Chip de Vídeo del 64
- Seguimos con los Vicassos
- ¿Riesgos? ¡No, gracias!
- Juegos
- Club Commodore
- Y todas vuestras colaboraciones

¡OS ESPERAMOS!

Commodore World es redactado en colaboración entre Microelectrónica y Control-Commodore y SIMSA

EQUIPO

Manuel AMADO; José Luis ERRAZQUIN; Miguel A. H. HIDALGO; María LOPEZ; Juan MARTINEZ; Pere MASATS; Jeffrey MILLS; Victoria MORALES; Fernando M. RODRIGUEZ; Diego ROMERO; Albert SANGLAS; Jordi SASTRE; Valerie SHANKS...
...Y NUESTROS LECTORES

SIMSA
Coordinador María López
Pedro Murugura, 4-8ºB — Madrid-16 — Tel.: (91) 259 54 78
Delegación en Barcelona:
Sant Gervasi de Cassoles, 39 despacho 4
Barcelona-22 — Tel.: (93) 212 73 45

MICROELECTRONICA
Y CONTROL-COMMODORE
Coordinador Pere Masats
Taquígrafo Serra, 7-5º
Barcelona-29
Tels.: (93) 250 51 03/02

Imprime: IBERDOS, S. A. Germán Pérez Carrasco, 24. Madrid-27
Depósito Legal: M-2944-1984

SUPER-INTERESANTISIMO

18 de junio

Gran Sorteo Commodore World

1 Viaje a Inglaterra

2 Sorteo amigos colaboradores

3 Amigos de RAMY Y ROMO

¡Ya! En nuestro último número decíamos que era muy posible que este mes tuviéramos ya nuestro suscriptor nº 5.000, ganador automáticamente del viaje a Estados Unidos. Pues así ha sido, por lo que ya nos encontramos listos para efectuar el sorteo del viaje a Inglaterra entre todos los demás suscriptores el mismo día que tendrá lugar el sorteo semetral de regalos entre nuestro gran equipo de colaboradores de programas y magia. Asimismo se sortearán los regalos entre los que nos han ayudado a buscar los nombres para los chavales de las aventuras de RAMY y ROMO.

LUGAR Y FECHA DEL SORTEO

El sorteo tendrá lugar el lunes día 18 de junio a las 7,30 de la tarde en los salones del Colegio English Montessori School, c/ Eduardo Vela, 10 - Aravaca Madrid-23.

La página 6 está completamente dedicada a la relación de los premios que se sortearán, las firmas que han contribuido en los concursos y los detalles sobre los dos viajes.

NUMERO DE SUSCRIPTOR

Habrá un pequeño número de nuevos suscriptores que observarán con cierto asombro que tienen un número de suscriptor inferior al 5.000. La razón de esta "anomalía" se debe a haber quedado "vacante" ese número por dos motivos principales: en primer lugar alguna baja que se ha producido de entre los sucriptores primitivos de Club Commodore y en segundo lugar porque algunos antiguos suscriptores no se dieron cuenta de que Commodore World era continuación de Club Commodore y... se suscribieron de nuevo... encontrándose con ¡dos suscripciones! No quisimos "recuperar" estos números vacantes durante la carrera hacia el 5.000 a fin de que no hubiera suspicacias de engaño... y... todos andamos ya un poco escarmentados de "inteligentes" y obscuras campañas de promoción y similares. Una vez que el 5.000 ha salido, y ya a nadie le afecta, hemos decidido recuperar estos números vacantes. Por otro lado, si esos números "recuperados" llegados después del 5.000 entran en el sorteo a Londres, no es justo que los que han llegado igualmente con horas de diferencia y que tengan el 5.001 ó 5.010, no puedan entrar. Por esta razón se han admitido a sorteo todos los llegados hasta hoy.

...Y PARA LOS SUSCRIPTORES DEL FUTURO.... ¿QUE?

Bueno, vale, tranquilos. Ya estamos pensando en algo para el número 8.000... pero... de momento tenemos la imaginación extenuada y tomándose un descanso. Sin embargo, prometemos que haremos algo que esté muy bien y hemos demostrado que lo que prometemos lo cumplimos (...excepto los repartos ocasionales de nuestras suscripciones en que tenemos que contar con la "INVALIOSA DESCOLABORACION" de nuestro incomparable servicio —es un decir— de CORREOS - pero, eso sí, número que nos reclaman, número que enviamos de nuevo) por cierto...

Convocatoria del Primer Concurso Nacional Escolar de Programación patrocinado por Commodore World y Microelectrónica y Control

Como ya anunciamos en nuestro número anterior publicamos hoy oficialmente la convocatoria de este concurso que pretendemos que sea un importante acontecimiento en el mundo de la informática española y que signifique un gran paso en su desarrollo.

BASES

1º Podrán participar en el mismo todos aquellos niños y jóvenes que estén en edad escolar pre-universitaria o equivalente.

2º Habrá tres grupos definidos:
a) E.G.B.
b) B.U.P.
c) F.P.

3º Los trabajos pueden ser presentados por equipos o por individuales.

4º Los trabajos presentados por equipos deben especificar:
a) Nombre del equipo.
b) Centro al que representan.
c) Nivel de enseñanza que cursan (E.G.B.-B.U.P.-F.P.).
d) Nombre del profesor, monitor o coordinador.

5º Los trabajos presentados por individuales deben ir acompañados de:
a) Nombre del autor.
b) Edad y curso que estudia.
c) (i) Si concursa representando a un centro: nombre del mismo y nombre del profesor responsable.
(ii) Si concursa por sí mismo: nombre del colegio donde estudia y firma del director o profesor responsable avalando que en tanto cuanto conoce al concursante el trabajo que presenta es suyo propio.

6º El tema del concurso es la realización de un programa educativo sobre materia libre a elegir por los concursantes. La extensión del programa es también de libre elección.

7º Cada centro puede presentar el número de equipos o concursantes individuales que desee, asimismo cada equipo o concursante puede enviar el número de trabajos que desee, siempre que cada trabajo sea enviado por separado.

8º Los trabajos deben incluir:
a) Nombre del programa.
b) Nombre del computador para el que está confeccionado y periféricos adicionales que se necesitan (si fuese el caso).
c) Objetivo del programa.
d) Explicación detallada del programa (a máquina o impresora).
e) Listado completo a impresora.
f) Grabación en cinta o disco.

9º Todos los trabajos presentados deben ser originales e inéditos.

10º Los centros que pueden presentar equipos o concursantes individuales son:
a) Todo colegio escuela o instituto, público o privado, que imparta las enseñanzas de E.G.B., B.U.P. o F.P.
b) Centros de Informática que tengan grupos de enseñanza específica para niños y jóvenes menores de 18 años.
c) Clubs de informática que mantengan actividades específicas para niños y jóvenes menores de 18 años.

11º Los trabajos deben ser enviados antes del 31 de diciembre de 1984.

12º El fallo del concurso se dará a conocer en la primera semana de marzo de 1985 en el lugar, fecha y hora que se **indicarán**.

13º Los trabajos premiados quedarán propiedad de los patrocinadores.

Quisiéramos poder publicar en este número, junto con las bases, el número y cuantía exacta de los premios, pues aunque podemos ya adelantar que serán importantes, tanto para los concursantes como para los centros presentadores, todavía no han sido totalmente determinados. Sin embargo, hemos decidido presentar ya las bases para que centros y concursantes puedan ya ir comenzando a prepararse.

...AYUDA E IDEAS...

Si a algún lector se le ocurre alguna forma de envío de la revista que no sea altamente gravosa y que ofrezca una alternativa a Correos le agradeceremos enormemente la sugerencia. Distribuidores - repartidores de provincias - etc., agradeceremos presupuestos.

NUMEROS ATRASADOS CLUB COMMODORE Y COMMODORE WORLD

Vamos a ver. CLUB COMMODORE, publicada por MICROELECTRONICA Y CONTROL, publicó 16 números del 0 al 15 entre octubre de 1982 y enero de 1984.

COMMODORE WORLD, publicada por **S.I.M.S.A., editorial totalmente independiente de MEC y de la casa Commodore**, comenzó a publicar su número 1 en febrero de 1984, tras haber llegado a un acuerdo











Nº 5.000 - 10 DIAS EN EE.UU.

JOAN PUJIULA - 18 AÑOS

Fue un día de excitación y nervios. Se estaban haciendo suscripciones en Informat, en El Corte Inglés, llegaban por teléfono a Madrid y a la oficina de Barcelona. Habíamos montado un complicado tinglado para estar todos en continuo contacto según se iban formalizando las suscripciones...nº 4.995... y llega el correo con... 17 nuevas suscripciones. ¿Cómo decidíamos quién de ese nuevo lote era el 5.000? Unica solución justa: echarlo a suertes. Nos fuimos todos a Informat a Barcelona. Avisamos a Microelectrónica y Control y allí, en el stand de Commodore World se efectuó el sorteo público: —"Joan Pujiula i Juncá— 18 años - Sant Jaume de Llierca (Gerona) - C-64."

Joan vive en un pueblo de 800 habitantes al pie de los Pirineos catalanes. Es hijo de agricultores y estudia 4º grado de F.P. rama de Electricidad en Olot, a 13 kilómetros de su pueblo y ayuda a su padre cuando tiene tiempo libre.

Cuando le llamamos estaba trabajando en el campo con su padre, hablamos con su hermana y le volvimos a llamar una hora más tarde. Pensaba que era una broma de su hermana, no se lo podía creer, pensaba que su suscripción habría llegado demasiado tarde... Cuando le preguntamos si prefería cambiar el viaje por una cantidad en metálico... ¡Ni hablar!

Nos fuimos a Sant Jaume de Llierca a conocer a Joan y su familia. No quieren oír hablar de cambiar el viaje por dinero. Lo más importante para ellos es la educación de los hijos. María, su hermana, estudia en la Universidad de Barcelona. El padre nos dice: "Es casi increíble, yo nunca tuve suerte en lotería ni cosas de esas, nunca me tocó nada. Esto es una oportunidad única para Joan". A Joan lo que le hace una gran ilusión es la visita a la Moss Technology, la fábrica de circuitos integrados de Commodore en Pennsylvania. Decimos que Diego va a ir con él y la madre nos pregunta sobre Diego — Al decir que tiene 24 años y comentar que es también la primera vez que va a América la madre exclama: "¡Dios bendito!, ¡qué harán esos dos allí¡" Le aseguramos que Diego es una persona muy seria y responsable, que habla perfectamente bien el inglés y que alguien estará allí esperándoles.

Cuando hablamos con Joan todavía pensábamos que el viaje sería de 1 semana. Desde entonces hemos recibido invitación de la revista RUN, con excursión en helicóptero y el viaje se ha extendido a 10 días. (Para mayores detalles sobre el viaje leer la página siguiente.)

Nos despedimos de la familia Pujiula con quien pasamos un día inolvidable en un bellísimo valle montañés.

---

con MEC para absorber dentro de sus páginas a Club Commodore y hacerse cargo, asimismo, de los suscriptores que tenía.

Aquellos antiguos suscriptores de **Club Commodore** a quienes les falten los números 13, 14 y/ó 15, por favor pedidlos a MEC en Barcelona, escribid a Pere Masats. Nosotros no los tenemos ni somos responsables de CLUB COMMODORE. Sólo contamos con un par de colecciones para referencia y poder prestar un servicio de **venta** de fotocopias para aquellos lectores que lo deseen.

Los suscriptores, antiguos o nuevos, que no hayan recibido algún número de **COMMODORE WORLD**, pedídnoslo, por favor a S.I.M.S.A. ya que MEC no los tiene ni es responsable de ellos.

Pidiendo cada cosa a su sitio hará que lo recibáis todo mejor y más rápidamente.

PRECIO DE COMMODORE WORLD

Finalmente, recordamos a todos que a partir del 1º de julio el precio de Commodore World aumentará en un 15%. El precio de venta al público será de 315 pesetas y el de la suscripción de 2.530 al año (11 números).



FIRMAS COLABORADORAS

COMMODORE BUSINESS MACHINES
COMPUTERWORLD/ESPAÑA
ENGLISH MONTESSORI SCHOOL
IBERCHARTER
INDESCOMP
MICROELECTRONICA Y CONTROL (MEC)
MICRO INFORMATICA Y SOFT (MISSA)
S.I.M.S.A. - COMMODORE WORLD
RUN (EE.UU.)

ESTADOS UNIDOS

El viaje a EE.UU. que inicialmente estaba programado para una semana ha crecido hasta cumplir 10 días. El viaje y alojamiento en Nuevo York estará a cargo de S.I.M.S.A. La Casa Commodore de EE.UU. ha invitado a Joan y Diego a pasar dos días en Filadelfia, incluyendo los viajes desde Nueva York y vuelta, para visitar la Fábrica de Ensamblaje de Commodore Business Machines y la Fábrica de Circuitos Integrados — Moss Technology, ambas en Pennsylvania. Asimismo, la revista RUN, hermana gemela de Commodore World en América, dedicada específicamente a los equipos Commodore, les ha invitado a pasar tres días en Boston, incluyendo los viajes, desde donde les trasladará en helicóptero a la base de la editorial en Petersborough.

Ibercharter, la agencia de viajes con quien se contrató el servicio, ha querido añadir su granito de arena invitando a Joan y Diego a un tour completo de un día del Alto y Bajo Manhattan.

VIAJE A INGLATERRA

El viaje de una semana a Inglaterra estará totalmente a cargo de Microelectrónica y Control. Una persona de MEC acompañará al ganador en este viaje. Desde Londres se trasladarán a Corby, en el norte del país para visitar las instalaciones de COMMODORE y la Fábrica de Ensamblaje.

Es una pena que por un mes hemos perdido el Commodore Computer Show que se celebra en Londres del 7 al 9 de junio, aunque siendo éste un importante acontecimiento anual, Carlos Domenech ya le está dando vueltas en la cabeza para organizar algo que permita llevar a nuestros lectores al Show del año que viene.

SORTEO ENTRE LOS COLABORADORES DE PROGRAMAS, MAGIA Y AMIGOS DE RAMY Y ROMO

S.I.M.S.A. sorteará tres premios de 15.000, 10.000 y 5.000 pesetas entre los commodorianos que han contribuido con programas y, otros tres de 3.000, 2.000 y 1.000 pesetas entre aquellos que nos han estado echando una mano a todos con trucos y recetas mágicas. Asimismo, **Microelectrónica y Control** dará tres premios para cada una de las dos categorías consistentes en soft y juegos, equivalentes en valor a los premios en metálico dados por S.I.M.S.A. Igualmente se sortearán dos reglas-reloj-calculadora y 6 juegos para cada categoría.

Pablo Nistal, ganador del concurso Amigos de RAMY y ROMO, ha recibido una regla-reloj-calculadora y un juego para su Commodore. Entre todos los demás commodorianos que han participado en la búsqueda de nombres para los amiguetes humanos de las memorias RAMY y ROMO se sortearán 1 regla-reloj-calculadora y un juego.

Las reglas han sido donadas por COMPUTERWORLD/ESPAÑA, 4 juegos por Microinformática y Soft MISSA y 10 POR INDESCOMP.

Deseamos expresar nuestro más sincero agradecimiento a las firmas que han contribuido al éxito de nuestros concursos y hacer una pequeña presentación de las mismas.

COMPUTERWORLD/ESPAÑA

COMPUTERWORLD/ESPAÑA, el único periódico de la informática en España, lanzó su primer número el 13 de mayo de 1981. Sale los miércoles de todas las semanas y solamente se puede adquirir por suscripción. El pasado mes de junio lanzó MicroSistemas dedicada a los ordenadores personales, mensual y con venta abierta al público o por suscripción. Pertenece a la cadena C.W. Communications, la primera compañía Internacional dedicada solamente a publicaciones de informática en 24 países. Tiene su sede en Madrid, Barquillo, 21 y oficinas en Gravina, 13 y en Barcelona, San Gervasio de Cassoles, 39.

ENGLISH MONTESSORI SCHOOL

El Colegio English Montessori fue fundado en 1973 por Milagros King, está reconocido por el Mº de Educación Español como colegio extranjero Inglés y por el Mº Británico de Educación por lo que sus estudios son automáticamente convalidados. Los estudios se pueden cursar desde la edad de Kindergorten. En este momento están montando un laboratorio de computadores que comenzará a funcionar en su primera fase en julio con cursos de verano y el próximo septiembre para todo el Colegio.

Tiene 3 centros en Madrid, uno en Aravaca en c/Eduardo Vela, 10 y dos en Chamartín, c/ Triana, 65 y Alfonso XIII, 40.

IBERCHARTER

Ibercharter es algo más que una agencia de viajes - Angel, uno de sus directores es capaz de organizar desde una excursión al Museo del Prado a una expedición al Everest y a los mejores precios posibles. Tiene su propia organización mayorista para Canarias, Baleares y Grecia y actúan como eficaz agencia para cualquier otra parte del mundo. Sus oficinas se encuentran en Madrid-9 - Alcalá, 181 y Barcelona-3 - Vía Layetana, 33.

INDESCOMP

Casi no es necesario hacer una presentación de INDESCOMP, una de las casas más prestigiosas y conocidas en la producción y venta de Sort y Hard para ordenadores personales y una de las pocas firmas españolas que exportan al extranjero, este tipo de productos, llegando a alcanzar el nº 1 de ventas en Inglaterra con el juego de 'La Pulga'. Es casi imposible poseer un Commodore o un Spectrum y no contar con alguna cinta de la casa Indescomp. Uno de sus más importantes productos, del que se sienten legítimamente orgullosos, es la placa que convierte el VIC, de un 22 columnas en un 40 × 80, colocándolo en una 1.ª posición en la familia de los personales. Tiene su sede en Madrid-16, en el Paseo de la Castellana, 179.

MICRO INFORMATICA Y SOFT (M.I.S.S.A.)

Compañía de reciente creación ha aparecido en el mercado con proyectos muy ambiciosos de traer a España los últimos y mejores productos de soft y hard para ordenadores personales. Su primer lanzamiento han sido los siempre esperados juegos para los Commodore. Queriendo poder ofrecer sus productos a los precios más competitivos, ha decidido lanzarlos por el sistema de Mail Order (Venta por Correo). Tiene su sede, inicialmente, en Madrid-16 - C/ Maestro Chapi, 15.

AMIGOS DE RAMY Y ROMO: KBIT Y KBITA

No podemos quejarnos de la gran respuesta que ha habido para encontrar nombre "informático" a los dos exploradores del mundo del computador. Ha habido para todos los gustos. El ganador lo hemos elegido por votación entre todo el equipo de redacción y casi por unanimidad: KBIT el niño y KBITA la niña cuyo padrino ha sido Pablo Nistal que vive en c/ El Empecinado, 46 de Móstoles. Entre todos los demás participantes, aspirantes a padrinos se sortearán 2 regalos (ver pág. 6).



METEDURAS DE PATA

Lamentamos que, en el número 4, correspondiente al mes de mayo, la imprenta "metiese la pata hasta el fondo" en dos anuncios.

*** En la página 61, en el anuncio de TRONIK, la línea "—Alquiler de cartuchos de juegos" debería estar colocada debajo de la línea "—Accesorios para el Commodore y VIC 20", y encima de la que dice "Libros y revistas".

*** En la página 67, en el anuncio de TECNHEL, la línea que dice "—Programa de XXXXXXXXXXXXX" la colocaron en imprenta y desconocemos los motivos que tuvieron para hacerlo.

Y la imprenta sigue empeñada en cambiar la famosa línea del programa para leer el directorio... Este programa que venía incompleto en el manual en español, que corregimos en el número 3 en la sección "correcciones del manual", que volvimos a corregir en el número 4, y que todavía no hemos conseguido que salga la línea bien impresa. Lo que en el número 3 apareció como espacios y en el 4 como paréntesis, realmente son los símbolos de "menor que" y "mayor que", es decir: <>

La línea sería:

530 IF A$<>CHR$(96) THEN IF A$<>CHR$(160) THEN B$=B$+A$

Esperamos que esta vez salgan los símbolos correctos.

Justo al cierre de la edición del número, nos han dicho que en algunos niveles de dificultad y si no le aciertas a las primeras bombas, en el programa "El bombardero loco" aparecido en el Video Casino del número 4, da error en la línea 157. Esto se puede solucionar aumentando el valor asignado a la variable BY en la línea 4, cambiando la línea por:

4 V=53248:POKEV+21,0:POKE2042, 13:BX=80:BY=60

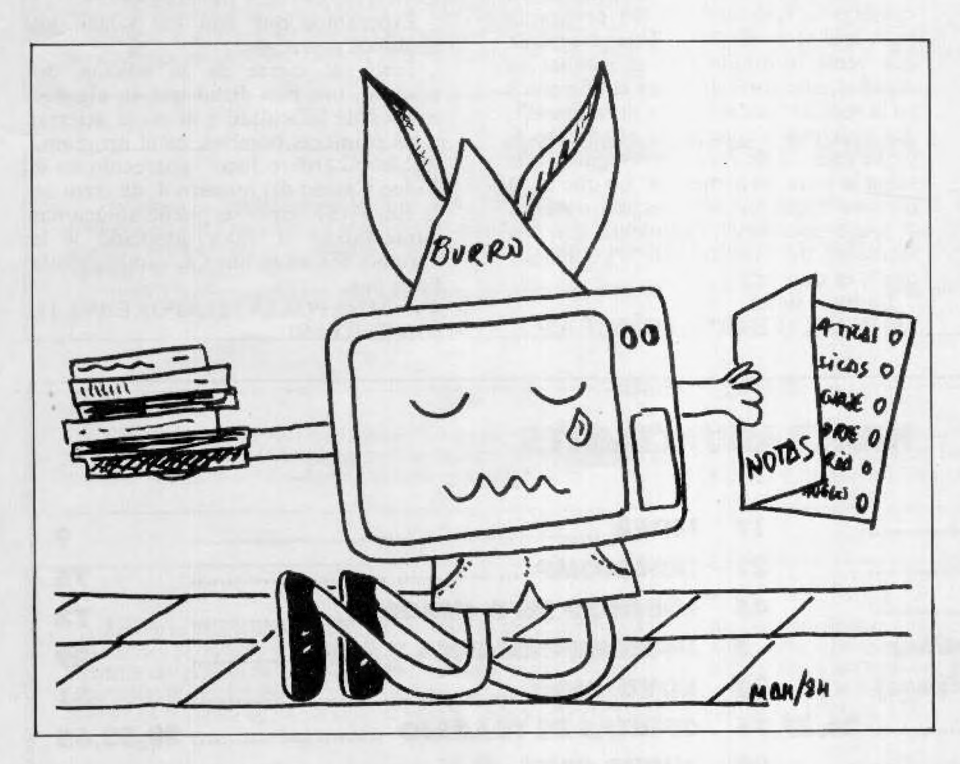
INDICE ANUNCIANTES

Programa de Evaluación para Colegios

Con la ayuda de este programa, el profesor más distraído podrá estar al tanto del Progreso de sus estudiantes. Este programa para llevar los datos estadísticos de la clase también puede ser modificado para manejar otras aplicaciones de datos numéricos.

Como profesor universitario, la parte más laboriosa de mi trabajo es estar al tanto del progreso (o falta de él) de los estudiantes pertenecientes a un curso determinado. El aspecto más pesado de este trabajo —leer y calificar exámenes, pruebas, deberes, etc.— también es la parte que más se resiste a automatizarse (a no ser que se utilicen exámenes tipo test que se corrigen por ordenador).

El ordenador más grande y más rápido del mundo no es capaz de leer y calificar un examen sobre "El Quijote" de Cervantes, una redacción sobre el sistema bancario de los Medici o un Problema de Cálculo. Sin embargo, los aspectos más mecánicos de un sistema de notas pueden ser enormemente simplificados mediante el uso del tipo de programa más adecuado.



C-64

El Programa "Lista de Clase" no solamente introduce, almacena y clasifica las notas de los estudiantes, sino que también asigna un nivel expresado en letras sobre la base de diversas entradas —exámenes, deberes, etc.— calificadas según una fórmula que proporciona unos resultados considerados "justos" sea cual sea la norma que se aplique. El programa permite que el usuario invente y modifique esta fórmula según sus necesidades.

Epoca de Exámenes

Además de manejar los datos básicos de controles y exámenes, el programa también tiene en cuenta los coeficientes multiplicadores para las notas; es decir, factores que pueden ser empleados para hacer un ajuste de acuerdo con la importancia relativa en la puntuación final. Por ejemplo, a lo mejor el profesor decide que la segunda evaluación a mitad de curso debe contar el doble de la primera, que el examen final debe contar 1 vez y 1/2 más que la segunda evaluación, y los deberes acumulados solamente la mitad de la primera evaluación. El programa toma los valores de dichos coeficientes multiplicadores como datos y los almacena junto con los datos de cada alumno. Estos coeficientes pueden ser reajustados o revisados en cualquier momento.

El programa también proporciona unos límites numéricos para la asignación de nivel expresado en letras. La fórmula elegida calcula la puntuación de cada estudiante y asigna un nivel basado sobre la información proporcionada por el profesor sobre el límite establecido (la A más baja, la B más alta, etc). Los límites también se almacenan con las notas y los coeficientes y pueden ser revisados con la misma facilidad.

Este programa está diseñado para incluir hasta 200 notas individuales.

Una opción de lista permite la representación y la impresión de una lista completa de estudiantes, junto con las puntuaciones hasta la fecha y el nivel expresado en letras, si este se desea. Las notas individuales, con todos los datos incluidos, pueden ser recuperados mediante una operación de "búsqueda" —esto también puede representarse en pantalla o imprimirse—.

También se pueden sacar resúmenes de los datos para toda la clase. Las opciones de resúmenes incluyen estadísticas (puntuación media y alumno medio y desviación standard), la districión de niveles (una gráfica de barra que muestra el número de estudiantes que han alcanzado un nivel determinado) y separación de niveles (las puntuaciones altas y bajas dentro de cada nivel).

El programa funciona en el Commodore 64 con una unidad de discos y utiliza cualquier impresora compatible. Sin embargo, la impresora no es imprescindible, dado que todas las rutinas de impresión son opcionales. Algunas de las representaciones en pantalla son multicolores. Pero un monitor en blanco y negro funciona perfectamente bien. La parte central de este programa ha sido adaptada del ejemplo de la lista de correos "Random File" de Commodore, presentada en el disco de demostración que acompaña la unidad de discos. Las subrutinas básicas de Leer y Escribir en Disco y el sistema de fichero índice son bastante parecidas. Sin embargo, las subrutinas especializadas opcionales que se incluyen aquí hacen que este programa resulte más complejo, con necesidad de un mínimo de 20K de RAM.

Empezando los deberes

Este programa necesita un disco formateado para el almacenamiento ▪

Programa de Evaluación para Colegios

```
1 REM ******************************
*
2 REM *
*
3 REM *      LISTA DE CLASE Y
*
4 REM *          NOTAS
*
5 REM *
*
6 REM ******************************
*
7 REM *
*
8 REM * CONVENIOS DE LISTADO EN LA
*
9 REM *   LINEA 7000
*
10 REM *
 *
11 REM ***************************
**
12 POKE53281,1:PRINT"[CLR]":PRINT"[BLK]
"
13 PRINT"[2CRSRD][SPC]LISTA[SPC]DE[SPC]
CLASE[SPC]Y[SPC]NOTAS":PRINT"[SPC](C
ON[SPC]OPCION[SPC]DE[SPC]NIVELES)"
14 PRINT:PRINT:PRINT:PRINT:PRINT"PUL
SAR[SPC]I[SPC]PARA[SPC]COMENZAR"
15 GETX$:IFX$<>"I"THEN15
19 DIMU(200),I$(200),HZ(2000):FD=0:F
X=5:CH=2:FP=1:PRINT"[CLR][2CRSRD]"
20 PRINT"[SPC]INSERTAR[SPC]DISCO[2SPC]
DE[SPC]LAS[SPC]NOTAS":PRINT:PRINT:PR
INT:PRINT:PRINT:PRINT
22 PRINT"[39SHIFTD]"
24 PRINT"[SPC]LO[SPC]HAS[SPC]HECHO?[2SPC]
S[SPC]PARA[SPC]CONTINUAR"
25 GETP$:IFP$<>"S"THEN25
26 PRINT"[CLR]":PRINT"[4CRSRD]":INPU
T"DESEAS[SPC]NIVELES[SPC](S/N)";GRD$

29 OPEN15,8,15,"I0":OPEN2,8,2,"#"
30 PRINT"[CLR][CRSRD][40SHIFTF]"
32 PRINT"[4SPC]OPERACION[SPC]FICHERO
[SPC]INDICE[2SPC]"
34 PRINT"[CRSRD][40SHIFTD]"
36 INPUT"[CRSRD][2SPC]LISTA[SPC]NUEV
A?[SPC]N[3CRSRL]";O$:PRINT""
37 IFO$="N"THEN2000
38 IFO$<>"S"THEN36
39 INPUT"[SPC]ESTAS[SPC]SEGURO";O$
40 IFO$<>"S"THENPRINT"[CLR]":GOTO36
43 GOTO2200
45 GOTO50
46 OPEN5,8,5,"0:INDEX,S,R"
47 FORI=1TO200:INPUT#5,I$(I)
49 NEXTI:CLOSE5
50 GOSUB1300
51 PRINT"[CLR]"
52 PRINT"[10SPC]MENU[SPC]DE[SPC]TRAB
AJO"
54 PRINT"[40SHIFTD]"
56 PRINT
58 PRINT"[6SPC]1=CREAR/ACTUALIZAR"
60 PRINT"[6SPC]2=BORRAR"
62 PRINT"[6SPC]3=BUSCAR"
63 PRINT"[6SPC]4=RESUMEN"
64 PRINT"[6SPC]0=FIN"
65 PRINT
66 INPUT"1[SPC],[SPC]2[SPC],[SPC]3[SPC]
,[SPC]4[SPC],[SPC]0[3SPC]1[3CRSRL]";
O$
68 IFO$="0"THEN2500
70 IFO$="1"THEN104
71 IFO$="4"THEN3000
72 IFO$="3"THEN300
74 IFO$<>"2"THEN50
76 GOTO200
100 REM ********************
102 REM *  CREAR FICHERO    *
103 REM ********************
104 PRINT"[CLR][2CRSRD][40SHIFTF]"
105 PRINT"[3SPC]CREAR/ACTUALIZAR[SPC]
FICHERO[4SPC]"
106 PRINT"[CRSRD][40SHIFTD]"
107 INPUT"NO.[SPC]ALUMNO[SPC]=0[3CRSRL]
";F
108 IFF=0THEN170
109 IFF>200THEN107
110 GOSUB1500
160 GOTO104
170 OPEN5,8,5,"@0:INDEX,S,W"
```

Pasa a siguiente página

de notas y un DISCO DISTINTO para cada clase. Después de cargar y ejecutar el programa, se pulsa la tecla I para empezar. Cuando el ordenador indique que se debe de insertar el disco de las notas en la unidad, el disco del programa debe ser sustituido por el disco utilizado para los ficheros. Para continuar la operación se pulsa la tecla S.

Primero el programa le pregunta al usuario si quiere la opción de niveles expresados en letras (se supone que esta operación es prematura al principio de curso), y luego pregunta si el disco de las notas está siendo utilizado por primera vez para esta clase específica. (Se puede utilizar un disco que contiene notas anteriores, como si fuese nuevo, dado que el programa no "verá" nunca las notas anteriores) no es necesario borrar el disco.

Si el disco se usa por primera vez, el programa pide que el usuario realice una doble verificación (para no borrar un disco por error) y pide a continuación que se inicie la entrada de coeficientes multiplicadores para los exámenes y los deberes. Dado que el programa, tal y como está escrito, maneja tres exámenes parciales, un examen final y un cálculo global de los deberes, se necesitan cinco de estos coeficientes. Estos pueden tener cualquier valor numérico, pero es aconsejable elegir valores que den una máxima puntuación teórica, para cualquier estudiante, de menos de 2.000 (Si se proyectan menos de tres exámenes parciales, o los deberes no se toman en cuenta, los coeficientes no utilizados son irrelevantes).

Una vez asignados los coeficientes, se solicitan límites para los niveles, si la opción de asignación por letras ha sido elegida. El límite de un nivel determinado significa la puntuación acumulada más baja que pueda

Viene de página anterior

```
175 FORI=1TO200:PRINT#5,I$(I);CHR$(1
3);
185 NEXT:CLOSE5
190 GOTO74
200 REM **********************
201 REM *   BORRAR FICHERO   *
202 REM **********************
210 PRINT"[CLR][2CRSRD][40SHIFTF]"
212 PRINT"[7SPC]BORRAR[SPC]EN[SPC]EL
[SPC]FICHERO[SPC]"
214 PRINT"[CRSRD][40SHIFTD]"
220 INPUT"NO.[SPC]ALUMNO[SPC]=[SPC]0
[3CRSRL]";F
230 IFF=0THEN260
233 IFF>200THEN220
235 IFI$(F)<>"1"THEN220
240 I$(F)="/":PRINT"[CRSRD]ALUMNO[SPC]
NO."F;"BORRADO"
250 GOTO220
260 OPEN5,8,5,"@0:INDEX,S,W"
265 FORI=1TO200:PRINT#5,I$(I);CHR$(1
3);
275 NEXT:CLOSE5
280 GOTO50
300 REM ***********************
301 REM *  BUSCAR/VISUALIZAR  *
302 REM ***********************
303 PRINT"[CLR][2CRSRD][40SHIFTF]"
304 PRINT"[15SPC]BUSCAR"
305 PRINT"[CRSRD][40SHIFTD]"
306 INPUT"LO[SPC]IMPRIMO[SPC](S/N)";
PO$
307 INPUT"NO.[SPC]ALUMNO[SPC]=[SPC]0
[3CRSRL]";F
308 IFF=0THENPO$="N"
309 IFF>200THEN307
310 IFF=0THEN50
315 IFI$(F)<>"1"THEN307
325 GOSUB5100
340 IFPO$="S"THENOPEN4,4
342 IFPO$="S"THENCMD4
360 PRINT"[3SPC]";SUR$"[SPC]"ID$
361 PRINT"[3SPC]";GIV$
362 PRINT:PRINT
363 PRINT"TEST[SPC]1[2SPC]"AXA
365 PRINT"TEST[SPC]2[2SPC]"BXB
366 PRINT"TEST[SPC]3[2SPC]"CXC
370 PRINT"FINAL[2SPC]"FEX
372 PRINT"T.[SPC]EN[SPC]CASA"HW
373 CUM=W1*AXA+W2*BXB+W3*CXC+WF*FEX+
WHW*HW
375 PRINT"NOTA[SPC]GLOBAL[SPC]";CUM
380 IFGRD$="S"THENPRINT"[SPC]RENDIMI
ENTO:[SPC]"G$
385 PRINT"------------------------"
:PRINT
386 IFPO$<>"S"THEN390
387 PRINT#4:CLOSE4
390 GOTO307
1100 REM *********************
1101 REM *NIVELES-RENDIMIENTO*
1102 REM *********************
1110 INPUT"[SPC]MENOR[SPC]D[2SPC]";L
D
1115 INPUT"[SPC]MENOR[SPC]C[2SPC]";L
C
1120 INPUT"[SPC]MENOR[SPC]C+[SPC]";L
P
1125 INPUT"[SPC]MENOR[SPC]B[2SPC]";L
B
1130 INPUT"[SPC]MENOR[SPC]B+[SPC]";L
Q
1135 INPUT"[SPC]MENOR[SPC]A[2SPC]";L
A
1140 RETURN
1300 REM ******************
1301 REM * LISTA DE CLASE *
1302 REM ******************
1303 PRINT"[CLR]":PRINT:PRINT:PRINT:
PRINT:PRINT"LISTA[SPC]?[SPC](S[SPC]O
R[SPC]N)"
1305 GETM$:IFM$=""THEN1305
1306 IFM$="N"THENRETURN
1307 IFM$="S"THEN1310
1308 GOTO1303
1310 PRINT:PRINT:PRINT"LO[SPC]IMPRIM
O?[SPC](PULSA[SPC]R[SPC]CUANDO[SPC]T
ERMINE)"
1311 GETZ$:IFZ$=""THEN1311
1312 IFZ$="N"THEN1324
1313 IFZ$="S"THEN1330
1314 GOTO1311
1316 REM
1324 PRINT:PRINT:PRINT"D[SPC]PARA[SPC]
VISUALIZAR":PRINT
1325 PRINT:PRINT"W[SPC]PARA[SPC]DETE
```

alcanzar dicho límite. De la forma en que está escrito el programa, los posibles niveles expresados en letras son F, D, C, C, B, B, y A: elegidas simplemente porque estos son los niveles oficiales en el centro donde yo trabajo.

Si se actualizan las notas establecidas, el flujo es semejante. El ordenador pregunta si el usuario quiere modificar los coeficientes y los límites previamente introducidos. Después de actualizar los coeficientes y los límites o si se ha decidido no realizar esta operación —se pasa a la rutina de "lista".

Cuando aparece "Lista?" en pantalla, el ordenador está preguntando si el usuario quiere la representación de toda la lista de la clase. Si no es necesario (esto se indica pulsando la tecla N), se pasa el menú principal. La tecla "S" presenta la opción de una copia impresa de la lista entera, pulsando a continuación la tecla R para pasar al menú principal.

Si no se necesita una copia impresa, se inicia la representación en pantalla pulsando la tecla D; la tecla W detiene la representación existente en pantalla. Después de fijar la representación, o después de representar la lista entera, se pulsa la tecla R para pasar al menú principal. El formato para la representación de la lista es una sola línea con el número de alumno, la puntuación acumulada hasta la fecha, el nivel en letras (si esta opción fue elegida al comienzo), los apellidos y el nombre.

Opciones de cursos

Existen cinco opciones en el menú principal: creación/actualización de ficha, borrar ficha, búsqueda, resumen y fin. La opción más importante, la de creación/actualización de fichero.

```
NERLA"
1326 PRINT:PRINT"R[SPC]PARA[SPC]ACTU
ALIZAR[SPC]O[SPC]SALIR"
1327 GETR$:IFR$<>"R"THEN1327
1330 FORF=1TO200
1332 IFI$(F)<>"1"THEN1355
1333 GOSUB5100
1334 CUM=W1*AXA+W2*BXB+W3*CXC+WF*FEX
+WHW*HW
1335 IFGRD$="S"THENGOSUB6000
1336 LE$="[SPC]"
1337 IFGRD$="S"THENLE$=G$
1338 IFZ$="S"THENOPEN4,4:CMD4
1340 PRINTTAB(2)FSPC(3)CUMSPC(2)LE$S
PC(3)SUR$"[SPC]"ID$",[SPC]"GIV$CHR$(
13)
1345 IFZ$="S"THENPRINT#4:CLOSE4:GOTO
1355
1349 FORZ=1TO300:NEXT
1350 GETE$:IFE$<>"W"THEN1355
1352 IFE$="W"THENGOTO1400
1355 NEXT
1400 GETQ$:IFQ$<>"R"THEN1400
1403 IFQ$="R"THEN1405
1405 SUR$="[SPC]":GIV$="[SPC]":ID$="
[SPC]"
1410 AXA=0:BXB=0:CXC=0:FEX=0:HW=0
1415 RETURN
1500 REM ************************
1501 REM * COMPRUEBA/ACTUALIZA *
1502 REM ************************
1510 IFI$(F)<>"1"THEN1600
1515 GOSUB5100
1520 PRINT"[RED]"
1521 PRINT"[CLR]"
1523 PRINTF"[3SPC]"SUR$"[2SPC]"ID$
1524 PRINT"[5SPC]"GIV$
1525 PRINT"TEST[SPC]1[2SPC]"AXA,"TES
T[SPC]2[2SPC]"BXB,"TEST[SPC]3[3SPC]"
CXC
1526 PRINT"[2SPC]FINAL[SPC]"FEX"[2SPC]
EN[SPC]CASA"HW
1529 PRINT:PRINT"[3SPC]*[SPC]=[SPC]N
O[SPC]CAMBIARLO[4SPC]"
1531 PRINT"--------------------------
-----"
1532 PRINT"[BLK]"
1535 INPUT"CAMBIARLO[SPC](S/N)[SPC]"
;UN$
1537 IFUN$<>"S"THENRETURN
1539 INPUT"APELLIDO[SPC]1";CS$
1541 IFCS$="*"THEN1545
1543 SUR$=CS$
1545 INPUT"APELLIDO[SPC]2";CG$
1547 IFCG$="*"THEN1551
1549 ID$=CG$
1551 INPUT"NOMBRE";CI$
1553 IFCI$="*"THEN1557
1555 GIV$=CI$
1557 INPUT"[SPC]TEST[SPC]1";AZ$
1559 IFAZ$="*"THEN1563
1561 AXA=VAL(AZ$)
1563 INPUT"[SPC]TEST[SPC]2";BZ$
1565 IFBZ$="*"THEN1569
1567 BXB=VAL(BZ$)
1569 INPUT"[SPC]TEST[SPC]3";CZ$
1571 IFCZ$="*"THEN1575
1573 CXC=VAL(CZ$)
1575 INPUT"[SPC]FINAL";CF$
1577 IFCF$="*"THEN1581
1579 FEX=VAL(CF$)
1581 INPUT"T.[SPC]EN[SPC]CASA";CW$
1583 IFCW$="*"THEN1586
1585 HW=VAL(CW$)
1586 GOSUB5200
1587 I$(F)="1"
1589 RETURN
1600 PRINT"[CLR]":PRINT"[4SPC]CREAND
O[SPC]NUEVO[SPC]REGISTRO"
1601 SUR$="[SPC]":GIV$="[SPC]":ID$="
[SPC]"
1602 AXA=0:BXB=0:CXC=0:FEX=0:HW=0
1605 PRINT:PRINT"ALUMNO[SPC]#"F:PRIN
T
1610 INPUT"APELLIDO[SPC]1";SUR$
1615 INPUT"APELLIDO[SPC]2";ID$
1620 INPUT"NOMBRE";GIV$
1625 INPUT"TEST[SPC]#1[5SPC]";AXA
1630 INPUT"TEST[SPC]#2[5SPC]";BXB
1635 INPUT"TEST[SPC]#3[5SPC]";CXC
1640 INPUT"FINAL[7SPC]";FEX
1645 INPUT"T.[SPC]EN[SPC]CASA[2SPC]"
;HW
1650 GOSUB5200
1655 I$(F)="1"
1660 RETURN
```

Pide que el usuario asigne un número del 1 al 200. Si el número corresponde a una ficha inactiva, registrada por el índice de ficheros, aparece el mensaje Creando Nuevo Registro, junto con el nombre de alumno, y se le pide al usuario los apellidos, el nombre y las notas de los exámenes, 1, 2, 3 y el final.

Si todos estos datos no son disponibles todavía, se pulsa la tecla "Return" cuando aparece el mensaje, estableciéndose el cero como un valor por omisión. Una vez introducido un registro nuevo, la información se escribe en disco y se solicita un número de alumno nuevo.

Cuando se asigna un número de registro activo para la creación/actualización del fichero, el proceso resulta un poco diferente. La información contenida en dicho registro aparece en pantalla y el ordenador pregunta si el usuario quiere actualizarlo. Si la respuesta es negativa, se vuelve a solicitar un número de alumno. Si es positiva, se solicita toda la información necesaria. Cuando todas las entradas quedan completadas, los nuevos datos se escriben en disco y se pide un nuevo número.

Finalmente, cuando se pide un número de registro y se contesta con un cero, el programa da por terminada la rutina de creación/actualización. Entonces el índice de ficheros en el disco se actualiza y se vuelve a la opción de "lista" que, al ser rechazada vuelve al menú principal. La opción de borrar solicita un número de alumno. Si se introduce un número activo, el proceso de borrado simplemente convierte la bandera de índice de dicho número de activo en inactivo. A continuación, se solicita otro número de alumno para repetir el proceso. Si se contesta con un cero se vuelve a la opción "lista".

La opción de búsqueda primero

Viene de página anterior

```
2000 REM *********************
2001 REM *  COEFICIENTES  Y   *
2002 REM *      NIVELES       *
2003 REM *********************
2005 F=300:GOSUB5330
2006 PRINT#15,"U1:";CH;FD;FT;FS
2007 PRINT#15,"B-P:",CH;FP
2008 GOSUB5270
2011 INPUT#CH,W1
2012 INPUT#CH,W2
2013 INPUT#CH,W3
2014 INPUT#CH,WF
2015 INPUT#CH,WHW
2018 PRINT"[CLR]":PRINT
2019 INPUT"MANTENER[SPC]COEFS.[SPC]S
/N";K$
2020 IFK$="S"THEN2070
2021 IFK$<>"N"THEN2018
2022 PRINT"[RED]"
2023 PRINT"TEST[SPC]1[4SPC]"W1:PRINT
"TEST[SPC]2[4SPC]"W2
2024 PRINT"TEST[SPC]3[4SPC]"W3:PRINT
"FINAL[5SPC]"WF
2026 PRINT"T.[SPC]EN[SPC]CASA"WHW
2028 PRINT:PRINT"[2SPC]--------------
-------------------[2SPC]"
2029 PRINT"[BLK]"
2030 INPUT"CAMBIAR[SPC]COEF.[SPC]S/N
";K$
2032 IFK$="N"THEN2070
2035 INPUT"TEST[SPC]1";W1
2036 INPUT"TEST[SPC]2";W2
2037 INPUT"TEST[SPC]3";W3
2038 INPUT"FINAL[SPC]";WF
2039 INPUT"T.[SPC]CASA[SPC]";WHW
2040 GOSUB2940
2070 IFGRD$<>"S"THEN46
2071 PRINT"[CLR]":PRINT"SI[SPC]MODIFI
CA[SPC]LIMITES[SPC]O[SPC]INTRODUCIR
LOS"
2072 PRINT"POR[SPC]PRIMERA[SPC]VEZ."

2075 INPUT"EN[SPC]CASO[SPC]CONTRARIO
[SPC]N";K$
2080 IFK$="S"THEN2085
2081 IFK$<>"N"THEN2071
2083 GOTO2700
2085 F=400:GOSUB5330
2090 PRINT#15,"U1:";CH;FD;FT;FS
2095 PRINT#15,"B-P:",CH;FP
2097 GOSUB5270
2100 INPUT#CH,LD
2102 INPUT#CH,LC
2104 INPUT#CH,LP
2106 INPUT#CH,LB
2108 INPUT#CH,LQ
2110 INPUT#CH,LA
2115 PRINT"[CLR]":PRINT"[RED]"
2120 PRINT"[3SPC]MINIMO[SPC]D[2SPC]"
LD
2121 PRINT"[3SPC]MINIMO[SPC]C[2SPC]"
LC
2122 PRINT"[3SPC]MINIMO[SPC]C+[SPC]"
LP
2123 PRINT"[3SPC]MINIMO[SPC]B[2SPC]"
LB
2124 PRINT"[3SPC]MINIMO[SPC]B+[SPC]"
LQ
2125 PRINT"[3SPC]MINIMO[SPC]A[2SPC]"
LA
2130 PRINT:PRINT"[2SPC]-------------
--------------------[2SPC]":PRINT"[BLK]
"
2135 INPUT"CAMBIAR[SPC]LOS[SPC]MINIM
OS[SPC](S/N)";K$
2136 IFK$="N"THEN46
2140 INPUT"NUEVO[SPC]MINIMO[SPC]D[SPC]
";LD
2141 INPUT"NUEVO[SPC]MINIMO[SPC]C[SPC]
";LC
2142 INPUT"NUEVO[SPC]MINIMO[SPC]C+";
LP
2143 INPUT"NUEVO[SPC]MINIMO[SPC]B[SPC]
";LB
2144 INPUT"NUEVO[SPC]MINIMO[SPC]B+";
LQ
2145 INPUT"NUEVO[SPC]MINIMO[SPC]A[SPC]
";LA
2150 GOSUB2940
2155 GOTO46
2200 REM *********************
2201 REM * NUEVA LISTA       *
2202 REM * COEFS. Y  LIMITES *
2203 REM *********************
2204 FORF=1TO200:I$(F)="/":NEXT
2205 F=300:GOSUB5330
2210 PRINT"[CLR]"
```

pregunta al usuario si desea utilizar la impresora; si es así, cada registro buscado en esta secuencia será impreso por completo. A continuación, se solicita un número. Cuando se asigna un fichero activo, se representa o se imprime el nombre, la puntuación de pruebas y deberes, la puntuación acumulada basada sobre los coeficientes actuales y, si se ha elegido la opción de asignación de niveles al comenzar el programa, este nivel expresado en letras. Al completarse una representación o una impresión el programa solicita que se inicie una operación de búsqueda en el fichero siguiente. Al pulsar la tecla cero se vuelve a la opción "lista".

Las notas

Al elegir la opción de resumen, aparecerá en pantalla el mensaje "Espere un poco". Es necesario esperar, ya que se están leyendo todos los datos de los ficheros activos y las puntuaciones acumuladas están siendo tabuladas. Al terminar este proceso, aparece el menú de resumen: 1) mediano, media y desviación estandard; 2) distribución por niveles; 3) separación de niveles; 0) volver a la opción "lista".

La opción 1 da otro mensaje de espera. Es así porque el cálculo del mediano de una colección no ordenada de cifras brutas requiere un proceso bastante largo. Finalmente, el mediano, la media y la desviación estandard de las puntuaciones aparecen en pantalla.

Recordamos que el mediano es la puntuación en medio de un grupo de puntuaciones. Si el número total de puntuaciones es par, con el resultado de que no queda una exactamente en medio, la puntuación mediana será el término medio de las dos puntuaciones que se encuentran a cada lado del

```
2215 PRINT:PRINT"ENTRA[SPC]COEFICIEN
TES[SPC]DE[SPC]EXAMENES":PRINT:PRINT

2220 INPUT"[2SPC]TEST[SPC]1[SPC]";W1

2221 INPUT"[2SPC]TEST[SPC]2[SPC]";W2

2222 INPUT"[2SPC]TEST[SPC]3[SPC]";W3
2223 INPUT"[2SPC]FINAL[2SPC]";WF
2224 INPUT"T.EN[SPC]CASA";WHW
2225 F=300:GOSUB5330
2230 GOSUB2840
2240 IFGRD$<>"S"THENLD=0:LC=0:LB=0:L
Q=0:LA=0:GOTO2270
2250 PRINT"[CLR]":PRINT
2255 PRINT"[2SPC]ENTRA[SPC]LOS[SPC]L
IMITES[SPC]DE[SPC]NIVELES[2SPC]":PRI
NT:PRINT
2260 INPUT"[SPC]MINIMO[SPC]D[SPC]";L
D
2261 INPUT"[SPC]MINIMO[SPC]C[SPC]";L
C
2262 INPUT"[SPC]MINIMO[SPC]C+";LP
2263 INPUT"[SPC]MINIMO[SPC]B[SPC]";L
B
2264 INPUT"[SPC]MINIMO[SPC]B+";LQ
2265 INPUT"[SPC]MINIMO[SPC]A[SPC]";L
A
2270 F=400:GOSUB5330
2280 GOSUB2940
2290 GOTO51
2500 REM CERRAR FICHEROS
2520 CLOSE2:CLOSE5:CLOSE15:END
2700 REM * LEE LIMITES DE ANTIGUA LI
STA *
2705 F=400:GOSUB5330
2710 PRINT#15,"U1:";CH;FD;FT;FS
2715 PRINT#15,"B-P:";CH;FP
2720 GOSUB5270
2725 INPUT#CH,LD
2727 INPUT#CH,LC
2729 INPUT#CH,LP
2731 INPUT#CH,LB
2733 INPUT#CH,LQ
2735 INPUT#CH,LA
2740 GOTO46
2840 REM ********************
2841 REM * COEFICIENTES-TEST *
2842 REM ********************
```

```
2843 F=300:GOSUB5330
2845 PRINT#15,"B-P:";CH;FP
2851 PRINT#CH,W1
2852 PRINT#CH,W2
2853 PRINT#CH,W3
2854 PRINT#CH,WF
2855 PRINT#CH,WHW
2860 PRINT#15,"U2:";CH;FD;FT;FS
2865 GOSUB5270
2870 RETURN
2940 REM ********************
2941 REM * GUARDA MINS-NIVEL *
2942 REM ********************
2943 F=400:GOSUB5330
2945 PRINT#15,"B-P:";CH;FP
2950 PRINT#CH,LD
2951 PRINT#CH,LC
2952 PRINT#CH,LP
2953 PRINT#CH,LB
2954 PRINT#CH,LQ
2955 PRINT#CH,LA
2960 PRINT#15,"U2:";CH,FD;FT;FS
2965 GOSUB5270
2970 RETURN
3000 REM ***********
3001 REM * RESUMEN *
3002 REM ***********
3006 KT=0:FORF=1TO200 U(F)=0 NEXT
3007 PRINT"[CLR]":PRINT:PRINT PRINT"
[SPC]ESPERE[SPC]UN[SPC]POCO!"
3008 FORF=1TO200
3010 IFI$(F)="1"THENKT=KT+1
3015 IFI$(F)="/"THEN3050
3020 GOSUB5100
3040 CUM=W1*AXA+W2*BXB+W3*CXC+WF*FEX
+WHW*HW
3045 U(F)=CUM
3050 NEXT
3059 PRINT"[CLR]"
3060 PRINT:PRINT"[SPC]1[SPC]=[SPC]ME
DIANO,[SPC]MEDIA,[SPC]DESV.[SPC]ESTD
."

3065 PRINT:PRINT"[SPC]2[SPC]=[SPC]DI
STRIBUCION[SPC]POR[SPC]NIVELES[SPC]"

3067 PRINT:PRINT"[SPC]3[SPC]=[SPC]SE
PARACION[SPC]DE[SPC]NIVELES[3SPC]"
3075 PRINT:PRINT"[SPC]0[SPC]=[SPC]VO
```

Pasa a siguiente página →

centro. Por otro lado, la puntuación media es simplemente el término medio de todas las puntuaciones. La comparación de la mediana y la media (muchas veces serán muy parecidas) da una idea de la forma en que las puntuaciones se distribuyen, a lo largo de la escala.

La desviación estandard es la raiz cuadrada de los términos medios de los cuadrados de las desviaciones individuales. (Una desviación individual es la diferencia entre una puntuación individual y la media). La desviación estandard da cierta medida de lo mucho o poco que las puntuaciones se agrupan alrededor de la media.

La elección de las opciones 2 ó 3 del menú de resumen significa que el usuario eligió inicialmente las asignaciones de los niveles por letras y que se introdujeron los límites de niveles. La opción 2 simplemente presenta el número total de niveles en cada categoría, de la A a la F. Los números se presentan en pantalla junto con una gráfica de barras multicolor (la A es roja, la F negra) que resume la misma información de forma visual.

La opción 3 presenta la puntuación acumulada más baja y la más alta existentes dentro de cada categoría de niveles. Esta opción ha sido incluida basándose en la suposición de que es preferible que el conjunto de puntuaciones acumuladas dentro de cada nivel se agrupe de forma más o menos clara. Por ejemplo, si la C más alta de la clase se difiere de la B más baja por uno o dos puntos, de digamos, 600, la operación de separación de grados le permite al usuario verlo en seguida.

Hay que tener en cuenta que se puede volver al menú de resumen pulsando la tecla R después de ejecutar cualquiera de estas tres operaciones.

Viene de página anterior

```
LVER[SPC]A[SPC]RUTINA-ACTUALIZAR"
3079 PRINT
3080 INPUT"1[SPC],[SPC]2[SPC],[SPC]3
[SPC],[SPC]0[2SPC]1[3CRSRL]";A$
3081 IFA$="1"THEN3100
3082 IFA$="2"THEN3300
3083 IFA$="3"THEN3500
3085 IFA$="0"THENCH=2:GOTO50
3090 GOTO3059
3100 REM *** ESTADISTICAS ***
3101 PRINT"[CLR]":PRINT:PRINT:PRINT"
[SPC]ESPERE[SPC]UN[SPC]POCO!"
3105 FORT=0TO2000:HZ(T)=0:NEXT
3107 FORF=1TO200
3110 IFI$(F)<>"1"THEN3115
3112 X=INT(2*U(F)):HZ(X)=HZ(X)+1
3115 NEXT
3120 HH=0
3122 FORJ=0TO2000
3125 HL=HH:HH=HH+HZ(J)
3127 IFHH>KT/2ANDHL<KT/2THEN3140
3130 IFHH=KT/2ANDHL<HHTHENMK=J
3133 IFHH>KT/2ANDHL=KT/2THEN3142
3135 NEXT
3140 MD=J/2:GOTO3165
3142 MD=(J+MK)/4:GOTO3165
3165 SM=0:FORV=1TO200
3167 SM=SM+U(V):NEXT
3170 MN=SM/KT
3175 DV=0:FORV=1TO200
3176 IFI$(V)<>"1"THEN3179
3177 DV=DV+((U(V)-MN)↑2)
3179 NEXT
3180 SD=(DV/KT)↑.5
3185 PRINT"[CLR]"
3190 PRINT:PRINT:PRINT"[3SPC]RESUMEN
[SPC]ESTADISTICO"
3200 PRINT:PRINT
3203 PRINT"[2SPC]MEDIANO[SPC]"MD:PRI
NT:PRINT
3205 PRINT"[2SPC]MEDIA[SPC]"MN
3207 PRINT:PRINT"[SPC]DESV.[SPC]ESTD
."SD
3210 PRINT:PRINT:PRINT:PRINT:PRINT:P
RINT:PRINT
3212 PRINT"[2SPC]PULSA[SPC]R[SPC]PAR
A[SPC]VOLVER[SPC]AL[SPC]MENU"
3215 GETR$:IFR$<>"R"THEN3215
3220 IFR$="R"THEN3059
3300 REM *** DISTRIBUCION/NIVELES **
*
3302 PRINT"[CLR]":PRINT:PRINT:PRINT"
[SPC]ESPERE[SPC]UN[SPC]POCO!"
3305 FF=0:DD=0:CC=0:PP=0:BB=0:QQ=0:A
A=0
3310 FORZ=1TO200:CUM=U(Z)
3312 IFI$(Z)<>"1"THEN3333
3315 GOSUB6000
3320 IFG$="F"THENFF=FF+1
3322 IFG$="D"THENDD=DD+1
3323 IFG$="C"THENCC=CC+1
3324 IFG$="C+"THENPP=PP+1
3326 IFG$="B"THENBB=BB+1
3328 IFG$="B+"THENQQ=QQ+1
3330 IFG$="A"THENAA=AA+1
3333 NEXT
3335 SC=1:IFKT<100THENSC=2
3340 PRINT"[CLR]"
3347 PRINT"[RED]":LTH=SC*AA:GOSUB340
0
3348 PRINT"TOTAL[SPC]A'S[2SPC]"AA:PR
INT"[CRSRU]"
3352 PRINT"[CYN]":LTH=SC*QQ:GOSUB340
0
3353 PRINT"TOTAL[SPC]B+'S[2SPC]"QQ:P
RINT"[CRSRU]"
3357 PRINT"[PUR]":LTH=SC*BB:GOSUB340
0
3358 PRINT"TOTAL[SPC]B'S[2SPC]"BB:PR
INT"[CRSRU]"
3362 PRINT"[GRN]":LTH=SC*PP:GOSUB340
0
3363 PRINT"TOTAL[SPC]C+'S[2SPC]"PP:P
RINT"[CRSRU]"
3367 PRINT"[BLU]":LTH=SC*CC:GOSUB340
0
3368 PRINT"TOTAL[SPC]C'S[2SPC]"CC:PR
INT"[CRSRU]"
3372 PRINT"[YEL]":LTH=SC*DD:GOSUB340
0
3373 PRINT"TOTAL[SPC]D'S[2SPC]"DD:PR
INT"[CRSRU]"
3377 PRINT"[BLK]":LTH=SC*FF:GOSUB340
0
3378 PRINT"TOTAL[SPC]F'S[2SPC]"FF:PR
```

Si el usuario no queda satisfecho con algún aspecto de la puntuación acumulada o la distribución de niveles, se puede intentar la modificación de la situación mediante el ajuste de las fórmulas (es decir, mediante la alteración de los factores de coeficientes y límites de niveles). Para hacer esto simplemente se sale del programa a través del menú principal (opción 0), se vuelve a ejecutar (RUN) y se introducen los nuevos valores de coeficientes y de límites cuando sean solicitados. Tales modificaciones se pueden realizar de forma sencilla y rápida, y se pueden probar muchas posibilidades para conseguir un resultado aceptable.

Modificaciones

Una vez que comprenda la lógica de este programa, el usuario podrá modificarlo, según sus propias necesidades. Por ejemplo, si el usuario enseña un curso en que los trabajos, en vez de los exámenes, sirven de criterio, podría ser necesario modificar las asignaciones en las diversas representaciones y poner "Trabajo" en vez de "Evaluación". (Recuerda que las calificaciones de los trabajos tendrán que ser numéricas, o por lo menos fáciles de convertir en números). A lo mejor quieres cambiar el número de exámenes, o incluso registrar cada asignación de deberes por separado. El diseño básico de este programa se prestará fácilmente a tales modificaciones.

El mismo principio se aplica si quieres utilizar otra graduación por letras (A-, B-, etc.) además de las incorporadas en el programa. Sin embargo, en este caso podrías preferir modificar el resumen de la distribución de grados para conseguir una representación visualmente atractiva.

```
INT"[CRSRU]"
3302 PRINT PRINT PRINT"[SPC]PULSA[SPC]
R[SPC]PARA[SPC]VOLVER[SPC]AL[SPC]MEN
U"
3302 GETR$ IFR$<>"R"THEN3302
3305 GOTO3059
3400 W=INT(LTH/2):E=2*(LTH-W)
3402 IFLTH=0THENPRINT"[BLK]" RETURN
3403 PRINT"[CRSRU]"
3405 FORI=1TOW PRINT"[RVSON][SPC][RVSOFF
"[illegible]
3410 IFE=1THENPRINT"[COMM&]",
3415 PRINT"[BLK]"
[illegible]
3500 REM *** SEPARACION/NIVEL ***
3502 PRINT"[CLR]":PRINT:PRINT:PRINT"
[SPC]ESPERE[SPC]UN[SPC]POCO!"
3503 AH=1016:QH=LA-1:BH=LQ-1:PH=LB-1
:CH=LP-1:DH=LC-1:FH=LD-1
3505 AG=LA:QG=LQ:BG=LB:PG=LP:CG=LC:D
G=LD:FG=0
3507 FORJ=1TO200
3510 CUM=U(J)
3515 IFI$(J)="/"THEN3560
3520 IFU(J)>LA-1ANDU(J)<AHTHENAH=U(J
)
3522 IFU(J)<LAANDU(J)>QGTHENQG=U(J)
3524 IFU(J)>LQ-1ANDU(J)<QHTHENQH=U(J
)
3526 IFU(J)<LQANDU(J)>BGTHENBG=U(J)
3528 IFU(J)>LB-1ANDU(J)<BHTHENBH=U(J
)
3530 IFU(J)<LBANDU(J)>PGTHENPG=U(J)
3532 IFU(J)>LP-1ANDU(J)<PHTHENPH=U(J
)
3534 IFU(J)<LPANDU(J)>CGTHENCG=U(J)
3536 IFU(J)>LC-1ANDU(J)<CHTHENCH=U(J
)
3538 IFU(J)<LCANDU(J)>DGTHENDG=U(J)
3540 IFU(J)>LD-1ANDU(J)<DHTHENDH=U(J
)
3542 IFU(J)<LDANDU(J)>FGTHENFG=U(J)
3560 NEXT
3570 VA=0:VQ=0:VB=0:VP=0:VC=0:VD=0:V
F=0
3580 FORJ=1TO200
3585 IFI$(J)="/"THEN3605
3588 CUM=U(J)
```

```
3590 GOSUB6000
3592 IFG$="A"THENVA=1
3594 IFG$="B+"THENVQ=1
3596 IFG$="B"THENVB=1
3598 IFG$="C+"THENVC=1
3600 IFG$="C"THENVC=1
3602 IFG$="D"THENVD=1
3604 IFG$="F"THENVF=1
3605 NEXT
3607 PRINT"[CLR]"
3610 IFVA=1THENPRINT"[SPC]MENOR[2SPC]
A[2SPC]"AH
3612 IFVQ=1THENPRINT"[SPC]MAYOR[2SPC]
B+[SPC]"QG
3615 IFVQ=1THENPRINT"[SPC]MENOR[2SPC]
B+[SPC]"QH
3617 IFVB=1THENPRINT"[SPC]MAYOR[2SPC]
B[2SPC]"BG
3620 IFVB=1THENPRINT"[SPC]MENOR[2SPC]
B[2SPC]"BH
3622 IFVP=1THENPRINT"[SPC]MAYOR[2SPC]
C+[SPC]"PG
3625 IFVP=1THENPRINT"[SPC]MENOR[2SPC]
C+[SPC]"PH
3627 IFVC=1THENPRINT"[SPC]MAYOR[2SPC]
C[2SPC]"CG
3630 IFVC=1THENPRINT"[SPC]MENOR[2SPC]
C[2SPC]"CH
3632 IFVD=1THENPRINT"[SPC]MAYOR[2SPC]
D[2SPC]"DG
3635 IFVD=1THENPRINT"[SPC]MENOR[2SPC]
D[2SPC]"DH
3640 IFVF=1THENPRINT"[SPC]MAYOR[2SPC]
F[2SPC]"FG
3670 PRINT PRINT"[SPC]PULSA[SPC]R[SPC]
PARA[SPC]VOLVER[SPC]AL[SPC]MENU"
3675 GETR$ IFR$<>"R"THEN3675
3680 GOTO3059
5100 REM *******************
5105 REM * FDD BLOCK READ *
5108 REM *******************
5110 GOSUB5300
5120 PRINT#15,"U1:",CH,FD,FT,FS
5130 PRINT#15,"B-P ",CH,FP
5140 GOSUB5270
5150 INPUT#CH,SUR$
5152 INPUT#CH,GIV$
```

Pasa a siguiente página

Si tienes un VIC-20 en vez de un C-64, la lógica del Basic funcionará fácilmente, con tal de que disponga de una expansión de memoria lo suficientemente amplia (recuerda que hacen falta más de 20K de espacio de trabajo). La única sentencia "Poke" del programa (línea 12) fija el color de pantalla en blanco, y para el VIC-20 sería POKE36879, 24. Sin embargo, dado que unas de las representaciones en pantalla se diseñaron pensando en la pantalla, de 40 columnas del C-64, podría ser necesario modificarlas para que queden bien en la pantalla del VIC-20.

CIVILMATICA 84

Las principales empresas de equipos en la Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos de Madrid.

El ciclo de conferencias técnicas fue ampliado con intervenciones de especialistas que explicaron las últimas realizaciones informáticas en estos aspectos de la Ingeniería.

Viene de página anterior

```
5153 INPUT#CH,ID$
5154 INPUT#CH,AX%
5155 INPUT#CH,BX%
5156 INPUT#CH,CX%
5157 INPUT#CH,FEX
5158 INPUT#CH,HW
5190 RETURN
5200 REM ******************
5201 REM * FDD BLOCK WRITE *
5202 REM ******************
5210 GOSUB5330
5220 PRINT#15,"B-P:";CH;FP
5230 PRINT#CH,SUR$
5231 PRINT#CH,GIV$
5232 PRINT#CH,ID$
5233 PRINT#CH,AX%
5234 PRINT#CH,BX%
5235 PRINT#CH,CX%
5236 PRINT#CH,FEX
5237 PRINT#CH,HW
5240 PRINT#15,"U2:";CH,FD;FT;FS
5250 GOSUB5270
5260 RETURN
5270 REM ***************
5271 REM * ERROR CHECK *
5272 REM ***************
5280 INPUT#15,EN,EM$,ET,ES
5290 IFEN=0THENRETURN
5300 PRINT"ERROR[SPC]STATUS:";EN;EM$
,ET;ES
5310 INPUT"CONTINUE?",Y$:IFY$="Y"THE
NRETURN
5320 STOP
5322 REM *********************
5324 REM * SET TRACK & SECTOR *
5326 REM *********************
5330 IFF<358THENF1=0:F2=22:F3=1:GOTO
5370
5340 IFF>357ANDF<472THENF1=357:F2=20
:F3=19:GOTO5370
5350 IFF>471ANDF<580THENF1=471:F2=19
:F3=25:GOTO5370
5360 IFF>579THENF1=579:F2=18:F3=31
5370 FT=INT(((F-F1)-1)/(F2-1))+F3
```

```
5380 FS=F-F1-(FT-F3)*F2+(FT-F3-1)
5390 RETURN
6000 REM ********************
6001 REM * COMP. DE NIVELES  *
6002 REM ********************
6020 G$="F"
6121 IFCUM>LD-1THENG$="D"
6122 IFCUM>LC-1THENG$="C"
6123 IFCUM>LP-1THENG$="C+"
6124 IFCUM>LB-1THENG$="B"
6125 IFCUM>LQ-1THENG$="B+"
6126 IFCUM>LA-1THENG$="A"
6130 RETURN
7000 REM ***************
7005 REM *  CLAVES DEL *
7010 REM *   LISTADO   *
7015 REM ***************
7020 REM "[CRSRU]" CURSOR UP
7025 REM "[CRSRD]" CURSOR DOWN
7030 REM "[CRSRL]" CURSOR LEFT
7035 REM "[CRSRR]" CURSOR RIGHT
7040 REM "[HOME]" CURSOR HOME
7045 REM "[CLR]" CLEAR SCREEN
7050 REM "[RVSON]" REVERSE ON
7055 REM "[RVSOFF]" REVERSE OFF
7060 REM "[SHIFT[SPC]SPACE]" SHIFTED
 SPACE
7065 REM "[BLK]" BLACK
7070 REM "[WHT]" WHITE
7075 REM "[RED]" RED
7080 REM "[CYN]" CYAN
7085 REM "[PUR]" PURPLE
7090 REM "[GRN]" GREEN
7095 REM "[BLU]" BLUE
7100 REM "[YEL]" YELLOW
7105 REM "[SHIFTD]" SHIFTED D
7110 REM "[SHIFTF]" SHIFTED F
```

casa de software s.a.

PARA COMMODORE 64

PRACTICALC 64



MANUAL EN CASTELLANO

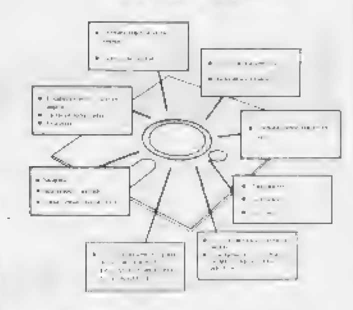
PROCESADOR DE TEXTO

35.000 caracteres, 240 columnas
Versión diskette: 21.500,–
Versión cartucho: 24.900,–
(grabación de documentos en diskette y cassette)

HOJA DE CALCULO

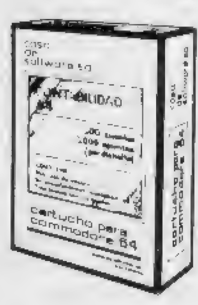
2.000 coordenadas
(funciones matemáticas, sort, gráficos...)
Versión cassette: 15.500,–
Versión diskette: 17.500,–

VIZAWRITE



MANUAL EN CASTELLANO

CONTABILIDAD 64 PROFESIONAL



300 cuentas
3.000 apuntes por disco
Listado de diario
Balance de sumas y saldos
Balance de situación
Extractos de cuenta
Listado de ficheros
Diario de cierre
Utilitarios, etc...

EQUIPO NECESARIO:
Ordenador: COMMODORE - 64
Unidad de disco: VC 1541
Impresora: VC 1525 ó MPS 801
Monitor ó T.V.

P.V.P. 24.550,–

ADAPTADO AL PLAN GENERAL CONTABLE ESPAÑOL

CONTABILIDAD DOMESTICA

1 cuenta ingresos
15 cuentas gastos
Listados por impresora y pantalla.
Análisis porcentual
Gráfico de gastos por pantalla.

VERSION CARTUCHO

P.V.P. 14.500,–

LAPIZ OPTICO Y SOFTWARE GRAFICO

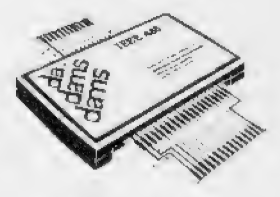


Dibujar en pantalla a mano alzada
formas geométricas, sombreados...
Grabación del dibujo en cassette.

P.V.P. 12.500,–

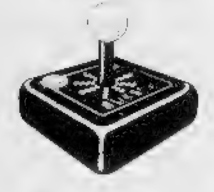
INTER FACE IEEE 488

Para conectar cualquier periférico con protocolo IEEE 488 (floppys 8050, 8250, impresoras 8023, 8024 etc.) al COMMODORE 64



P.V.P. 29.000,–

JOY STICK DE PRECISION



Pulsador sensible
Cable extra largo
Para CBM 64, VIC 20, ATARI CX 2600

P.V.P. 2.950,–

CONTABILIDAD DOMESTICA

22 cuentas
Listado por pantalla o impresora de los ingresos y gastos de cada período definido.

VERSION CASSETTE para
C-64: 3.500,–
VIC-20: 2.500,–

BOLETIN DE PEDIDO

Nombre y dirección:

☐ Contabilidad 64 (profesional)
☐ Contabilidad doméstica (cartucho)
☐ Contabilidad doméstica (cassette-C 64)
☐ Contabilidad doméstica (cassette-VIC 20)
☐ Practicalc ☐ Vizawrite
☐ Joystick ☐ Lápiz óptico
☐ Interface IEEE 488
☐ Información detallada

FORMA DE PAGO
☐ Adjunto talón (añadir 250 pts. por producto para gastos de envío)
☐ Contra reembolso

Enviar a: **CASA DE SOFTWARE S.A. Aragón, 272, 8.º 6.º Barcelona-7** tel. 215 69 52

FICHEROS CBM

Indexados Secuenciales

4.—Ficheros Indexados Secuenciales en Discos CBM (ISAM): (Continuación)

Por Manuel AMADO

d) Programa:

Este artículo es continuación del anterior de la serie, y, como lo prometido es deuda, en este artículo voy a describiros y mostraros un sencillo programa ejemplo del algoritmo gestor de ficheros indexados secuenciales, descrito en el anterior artículo (os aconsejo que tengáis al lado el algoritmo mencionado cuando procedáis a la lectura del programa).

Antes de proceder a la descripción del programa, me gustaría remarcar que éste pretende ser simplemente un ejemplo del manejo de una serie de rutinas que permiten el mantenimiento de un ISAM.

Normalmente, para la instalación de la mayoría de las diferentes estructuras de ficheros, se realizan dos procesos perfectamente diferenciados:

1) creación del fichero
2) gestión del fichero

En este caso, estos dos procesos corresponden a dos programas diferentes, llamados CREAISAM e ISAM respectivamente. Veamos cada uno de ellos:

1) CREACION DEL FICHERO

Antes de proceder a trabajar con el fichero que deseamos procesar, hay



que definir los parámetros principales de éste y crear los ficheros físicos correspondientes en el disco, de forma que quede reservado el espacio en disco necesario.

En el presente caso, necesitamos crear dos ficheros en disco, el fichero índice, de tipo secuencial, y el fichero final de datos, de tipo relativo. El programa es el correspondiente al listado 1.

—Creación del fichero índice:
(líneas 20-50)

En este caso, la creación es sinónimo de inicialización, pues además hay que INICIALIZAR el contenido de este fichero de modo que posibilite el posterior correcto funcionamiento el programa gestor (línea 50). Dadas las características de la rutina de búsqueda e inserción de nuevas claves en el fichero índice, se inicializan, mediante un bucle hasta el número máximo de registros que puede tener el fichero, las claves del índice a un contenido alfabéticamente mayor que cualquier clave que se pueda insertar en dicho índice, en este caso rellenándose cada clave con chr$(255) (clave falsa), con lo cual nos aseguramos que cualquier posición no ocupada estará al final del índice.

Y para evitar el tener que llevar un control externo de cuáles son los registros libres y ocupados del fichero de datos, se asocia a cada clave "falsa", como número de registro el valor del contador en aquel momento.

—Creación del fichero de datos:
(línea 60).

Esta operación también depende mucho del tipo de fichero físico que usemos. En el caso de un fichero de acceso directo, se trataría de reservar el área del disco que necesitamos mediante un bucle de "B-A". En el caso que estamos considerando, al tratarse de un fichero relativo, se trata de abrir el fichero relativo en modo creación, para que se genere la correspondiente entrada en directorio, asignando una longitud de registro de 254 (DOPEN#3, ("@"+NF$),

8000
700
BASIC 4

D0, L254) para evitar así en el posterior programa gestor el tener que controlar las longitudes de cada campo, dado que la finalidad básica de este programa es mostrar cómo se realizan y gestionan ficheros ISAM. En el caso de que se vaya a realizar una gestión eficiente y correcta del fichero de datos, hay que fijar una longitud para cada campo, y escribir los diferentes campos en el disco FORMATADOS a su longitud máxima (rellenados o cortados según sobrepase o no alcance dicha longitud con CHR$(32)). A continuación, se escribe en el último registro para que el DOS cree los sectores índice y reserve todos los bloques correspondientes. Para más detalles sobre el manejo de ficheros relativos, os recomiendo la lectura de la serie dedicada a este tipo de ficheros que apareció en la revista CLUB COMMODORE núm. 9-11, de JORDI SASTRE.

2) GESTION DEL FICHERO:

Este programa ha de realizar los procesos que se realizan en cualquier fichero, que son:

- —altas de nuevos registros
- —bajas de registros ya existentes
- —consulta y modificación de un registro.

En el programa gestor (listado 2), se ha obviado el proceso de modificación de un registro, dado que el objeto principal de este programa y del artículo es el mostrar las rutinas de gestión de un ISAM. En este caso, para realizar la modificación de un registro, bastaría con realizar una consulta, proceso descrito en el programa, y luego pedir si se desean o no modificar los campos del registro, a excepción del primero que es el que se ha tomado como clave de acceso o índice, y que no se puede modificar, puesto que entendemos por modificación el modificar el contenido del registro sin cambiar la clave de acceso. En el caso de querer cambiar también la clave de acceso, deberá realizarse una baja y una alta consecutivamente.

El programa se ha dividido en dos partes, un área principal, líneas 1000-2010, y un área de subrutinas, líneas 39-395. La parte IMPORTANTE del programa, es el conjunto de las subrutinas correspondientes a las líneas 39-310, en donde están todas las funciones necesarias para el mantenimiento del fichero, y que serán las que se aprovecharán luego al utilizar este fichero en cualquier programa. Esta parte está profundamente comentada en el algoritmo aparecido en el anterior artículo, y ahora pasaremos a ver el campo principal del programa:

```
PROGRAMA: CREAISAM
    0 CLR
    1 REM DSAVE"@CREAISAM"
    2 REM MANUEL AMADO / 21/MARZO/84
    10 REM DECLARACION DE PARAMETROS DEL ISA
    M
    20 NM=200 : NC=5 : DIM CL$(NM,2),R$(NC)
    30 NF$="INDEXADO" : RT$=CHR$(13)
    40 DOPEN#3,("@"+NF$+".C"),D0,W:PRINT#3,"
    0"
    50 FORI=1TONM:PRINT#3,CHR$(255)RT$STR$(I
    ):NEXT:DCLOSE#3
    55 SCRATCH(NF$),D0
    60 DOPEN#3,("@"+NF$),D0,L254:RECORD#3,(N
    M):PRINT#3,"A":RECORD#3,(NM):DCLOSE#3
```

```
PROGRAMA: ISAM
    0 CLR
    1 REM DSAVE"@ISAM"
    2 REM MANUEL AMADO / 06/ABRIL/84
    10 REM DECLARACION DE PARAMETROS DEL ISA
    M
    20 NM=200 : NC=5 : DIM CL$(NM,2),R$(NC)
    25 M1$="CLAVE[1SPC]EXISTENTE":M2$="CLAVE
    [1SPC]INEXISTENTE"
    30 NF$="INDEXADO" : RT$=CHR$(13) : GOTO
    1000
    38 :
    39 REM LECTURA DE LAS CLAVES DEL DISCO
    40 DOPEN#3,(NF$+".C"),D0 : INPUT#3,A$ :
    NT=VAL(A$): REM N.REGISTROS ACTUALES
    50 FORI=1TONM:INPUT#3,CL$(I,1),CL$(I,2):
    NEXT:DCLOSE#3
    55 DOPEN#1,(NF$),D0:RETURN
    58 :
    59 REM GRABACION DE LAS CLAVES EN EL DIS
    CO
    60 DOPEN#3,("@"+NF$+".C"),D0,W:PRINT#3,S
    TR$(NT)
    70 FORI=1TONM:PRINT#3,CL$(I,1)RT$CL$(I,2
    ):NEXT:DCLOSE#3:RETURN
    78 :
    79 REM LECTURA DEL REGISTRO NR
    80 RECORD#1,(NR):FORI=1TONC:INPUT#1,R$(I
    ):NEXT:RETURN
    88 :
    89 REM ESCRITURA DEL REGISTRO NR
    90 RECORD#1,(NR):R$="":FORI=1TONC:R$=R$+
    R$(I)+RT$:NEXT
    100 PRINT#1,R$;:RECORD#1,(NR):RETURN
    107 :
    108 REM SUBRUTINA DE BUSQUEDA DICOTOMICA
     DE LA CLAVE. EN CL$ LA CLAVE
    109 REM A BUSCAR. DEVUELVE EN T LA POSIC
    ION SI EXISTE. T=0 SI NO EXISTE
    110 M=INT(NT/2):J=0:IE=0
    120 IF CL$(M,1)>CL$ THEN M=INT(M-((M-J)/
    2)) : GOTO120
    130 J1=J:J=M:M=INT(M+((M-J1)/2)):IF M<>J
     GOTO120
```

LINEAS 20-30: Parámetros.
NM=Número máximo de registros del fichero. Definido en el programa de creación.
NC=Número de campos del fichero. Definido en el programa de creación.

Matrices principales:
CL$(I,1)=Clave índice I del fichero
CL$(I,2)=Número de registro de la clave I.
LINEA 1000-1060: Carga índice.

En la línea 1000 se carga el fichero índice. NT=Número de registros ocupados. (GOSUB40)
LINEAS 1005-1060: Menú del programa.

En él se selecciona la operación que se desea realizar en el fichero. A destacar que además de las tres operaciones básicas mencionadas anteriormente, también se ha incluido la opción de REGENERADO del fichero, que más adelante comentaré. En la opción de salida (línea 1055) se cierran todos los ficheros abiertos (DCLOSE), y se ejecuta un reseteado de la CPU (SYS 64790 para el 8000).
LINEAS 1100-1175: Altas

Se ejecuta la entrada e inserción de un nuevo registro.

a)entrada de la clave:

El poner a continuación del texto que acompaña al INPUT tres CHR$ (160) (SHIFT-SPACE) y tres cursor izquierda (CRSRL), es un sencillo truco para evitar que al contestar a un INPUT con un RETURN directo, sin haber entrado datos, se interrumpa la ejecución del programa, pues de este modo se consigue que siempre haya al menos un CHR$ (160) en el buffer del teclado. A continuación se testea si se ha entrado o no algún dato. En este caso, entrada de la clave, si se ha pulsado solamente RETURN, se graba el fichero índice en disco (GOSUB 60), y se vuelve al menú general. Búsqueda de la posición de inserción de la nueva clave (GOSUB 110). Si existe se vuelve a pedir clave. Con ello se evita la entrada de claves iguales. En el caso de desear la posibilidad de repetición de claves, simplemente eliminar la comparación (línea 1120)

b) entrada y grabación del registro

A continuación se entra el resto de los campos del registro (recordemos que en este fichero el primer campo es la clave), líneas 1140-1700. Se va a buscar el primer registro libre. (GOSUB 300), se inserta la clave (GOSUB 180), y se graba el registro.
LINEAS 1200-1290: Bajas

Si la clave no se ha encontrado, se manda el mensaje pertinente (línea

```
140 IF CL$(M+1,1)<CL$ AND M+1<=NT THEN M
=M+1  : GOTO120
150 T=M:RETURN
177 :
178 REM INSERCION DE UNA NUEVA CLAVE EN
CL$(I,J) EN LA POSICION T. CLAVE
179 REM EN CL$ : REGISTRO LIBRE EN NR
180 IF T=NT OR NT=0  THEN 210
190 IFT+1=NTTHENCL$(T+2,1)=CL$(T+1,1):CL
$(T+2,2)=CL$(T+1,2):GOTO210
200 FORIX=NTTOT+1STEP-1:CL$(IX+1,1)=CL$(
IX,1):CL$(IX+1,2)=CL$(IX,2):NEXTIX
210 CL$(T+1,1)=CL$:CL$(T+1,2)=STR$(NR):N
T=NT+1:RETURN
218 :
220 REM BORRADO DE UNA CLAVE EN CL$(I,J)
, EN LA POSICION T
230 FORIX=T+1TONT:CL$(IX-1,1)=CL$(IX,1):
CL$(IX-1,2)=CL$(IX,2):NEXT
240 RETURN
258 :
259 REM RUTINA DE VISUALIZACION DE PANTA
LLA DE TRABAJO
260 PRINT"[2HOME][1CLR][2CRSRD]"SPC(20)"
[1RVSON][5SPC]"+TT$+"[5SPC][1RVSOF]"
275 POKE224,8:RETURN
298 :
299 REM RUTINA DE BUSQUEDA DEL PRIMER RE
GISTRO LIBRE
300 IFNT>NMTHENRETURN
310 NR=VAL(CL$(NT+1,2)):RETURN
348 :
349 REM RUTINA DE VISUALIZACION DEL REGI
STRO LEIDO
350 PRINT"NOMBRE:[4SPC]"R$(1)
360 PRINT"DIRECCION:[6SPC]"R$(2)
370 PRINT"POBLACION:[2SHFSPC][4SPC]"R$(3)
380 PRINT"PROFESION:[6SPC]"R$(4)
390 PRINT"TELEFONO:[6SPC]"R$(5)
395 RETURN
998 :
999 REM CUERPO PRINCIPAL DEL PROGRAMA
1000 GOSUB 40
1005 PRINT"[2HOME][1CLR][2CRSRD]"SPC(20)
"[1RVSON][5SPC]MENU[1SPC]DE[1SPC]OPERACI
ONES[5SPC][1RVSOF]"
1010 PRINT"[2CRSRD][11CRSRR][1RVSON][1SP
C]1[1SPC][1RVSOF][4SPC]ALTAS
1020 PRINT"[2CRSRD][11CRSRR][1RVSON][1SP
C]2[1SPC][1RVSOF][4SPC]BAJAS
1030 PRINT"[2CRSRD][11CRSRR][1RVSON][1SP
C]3[1SPC][1RVSOF][4SPC]CONSULTAS
1035 PRINT"[2CRSRD][11CRSRR][1RVSON][1SP
C]4[1SPC][1RVSOF][4SPC]REGENERACION
1037 PRINT"[2CRSRD][11CRSRR][1RVSON][1SP
C]5[1SPC][1RVSOF][4SPC]SALIDA
1038 :
1039 REM SELECCIONA OPERACION A REALIZAR
1040 GETA$:IFA$=""THEN1040
1050 A=VAL(A$+CHR$(0)):IFA<1ORA>5GOTO104
0
1055 IFA=5THENDCLOSE:SYS64790
 : REM FIN
1060 ONAGOTO1100,1200,1300,1400
```

```
1098 :
1099 REM ALTAS
1100 TT$="ALTAS":GOSUB260
1110 INPUT"[1HOME][1CLR]NOMBRE[3SHFSPC][
3CRSRL]";CL$:IFASC(CL$)=160 THENGOSUB60:
GOTO1005
1115 GOSUB110
1120 IFLEFT$(CL$(T,1),LEN(CL$))=CL$THENP
RINT"[1RVSON]"+M1$:FORI=1TO1000:NEXT:GOT
O1110
1140 INPUT"[1CRSRD]DIRECCION[3SHFSPC][3C
RSRL]";R$(2):IFASC(R$(2))=160GOTO1140
1150 INPUT"[1CRSRD]POBLACION[3SHFSPC][3C
RSRL]";R$(3):IFASC(R$(3))=160GOTO1150
1160 INPUT"[1CRSRD]PROFESION[3SHFSPC][3C
RSRL]";R$(4):IFASC(R$(4))=160GOTO1160
1170 INPUT"[1CRSRD]TELEFONO[3SHFSPC][3CR
SRL]";R$(5):IFASC(R$(5))=160GOTO1170
1175 R$(1)=CL$:GOSUB300:GOSUB180:GOSUB90
:GOTO1110
1198 :
1199 REM BAJAS
1200 TT$="BAJAS":GOSUB260
1210 INPUT"[1HOME][1CLR]NOMBRE[3SHFSPC][
3CRSRL]";CL$:IFASC(CL$)=160 THENGOSUB60:
GOTO1005
1215 GOSUB110
1220 IFT>NTTHENT=NT
1222 IFCL$(T,1)<>CL$THENPRINT"[1RVSON]"+
M2$:FORI=1TO1000:NEXT:GOTO1200
1225 NR=VAL(CL$(T,2)):GOSUB80:GOUB350
1275 PRINT"ESTA[1SPC]SEGURO[1SHFSPC]<S>[
1SPC]/[1SPC]<N>[1SHFSPC]?"
1280 GETA$:IFA$<>"S"ANDA$<>"N"THEN1280
1285 IFA$="S"THENGOSUB220:CL$(NT,1)=CHR$
(255):CL$(NT,2)=STR$(NR)
1290 NT=NT-1:GOTO1210
1298 :
1299 REM CONSULTAS
1300 TT$="CONSULTAS":GOSUB260
1310 INPUT"[1HOME][1CLR]NOMBRE[3SHFSPC][
3CRSRL]";CL$:IFASC(CL$)=160 THENGOSUB60:
GOTO1005
1315 GOSUB110:IFCL$(T,1)=CL$THEN1320
1317 T=T+1
1320 IFT>NTTHENT=NT
1325 NR=VAL(CL$(T,2)):GOSUB80
1330 GOSUB350:PRINT"[2CRSRD][1RVSON][1SP
C]PULSE[1SPC]CUALQUIER[1SPC]TECLA"
1375 GETA$:IFA$=""GOTO1375
1380 GOTO1310
1398 :
1399 REM REGENERACION INDICE
1400 TT$="REGENERACION":GOSUB260
1405 INPUT"[1CLR]NUMERO[1SPC]DE[1SPC]REG
ISTROS[1SPC]TOTAL[1SPC]DEL[1SPC]FICHERO[
3SHFSPC][3CRSRL]";NR$
1410 POKE224,24:PRINT"[1HOME][1CLR]"
1415 NT=1:FORY=1TOVAL(NR$):NR=Y:GOSUB80
1420 CL$=R$(1):GOSUB110:PRINTCL$
1430 GOSUB180:NEXT:GOSUB60:GOTO1005
2000 FORNR=1TONM:GOSUB80
2010 CL$=R$(1):PRINT"[1HOME][1CLR]"CL$:G
OSUB110:GOSUB300:GOSUB180:NEXT:GOSUB60:D
CLOSE
```

1222). Se visualiza el contenido del registro (lín. 1225), y se pide conformidad. En caso afirmativo, se elimina la clave del índice (gosub 220), y se inserta la clave falsa de chr$(255) al final del índice ocupado, reservado el NR liberado (CL$ (NT,1)=CHR$(255): CL$ (NT,2)=NR).

LINEAS 1300-1380: Consultas.

La mecánica es parecida al anterior, pero con la excepción de que hay que realizar un ajuste en el índice T, que indica la posición de la clave buscada, para permitir búsquedas genéricas, (líneas 1315-1320).

LINEAS 1400-1430: Regeneración del índice.

Un detalle importante de este fichero es la seguridad de los datos en disco. Al formar parte la clave como un campo más del fichero de datos, ante cualquier pérdida de la información almacenada en ram (cortes de luz, fluctuaciones en la tensión, etc.) y que afectaría principalmente al índice, que hasta que no se finaliza una de las operaciones, (bajas o altas), no se graba en el disco. Ello conllevaría una pérdida de información que en muchos casos sería sensiblemente importante, sobre todo si nos hallásemos en medio de un proceso de entrada de datos, con lo cual se perderían todos los datos entrados hasta el momento. Pero como los datos están TODOS grabados en disco, mediante un sencillo proceso, un simple bucle, se puede REGENERAR el índice que se hubiese podido estropear.

Esto es lo que hace el presente apartado. Se pregunta primero al usuario cuántos registros cree que están ocupados, y a continuación realiza un barrido del fichero relativo según el número de registros indicado. Se va recogiendo el contenido del primer campo de cada registro (lín. 1415-1420), y se inserta dicha clave y su NR asociado en la matriz índice (GOSUB 110-GOSUB 180). Finalizado el bucle, se graba la matriz regenerada en disco (GOSUB 60), y se vuelve al menú. Finalmente, para finalizar el artículo, reseñaros a los que no sois usuarios del equipo 8000, que la instrucción POKE 224,X lo que realiza es definir la línea superior de la pantalla activa, de forma que la porción de la pantalla que queda disponible una vez ejecutada dicha instrucción es la comprendida entre la línea X y la 25 (el sistema entiende ahora que la pantalla EMPIEZA en la línea X y no en la 25). Este límite queda suprimido ejecutando dos "HOME" seguidos, o sea PRINT "[HOME] [HOME]". ■

Cuando se es COMMODORE es muy difícil ser modesto

SLOGAN

COMMODORE 64

Cuando se tiene 64 K de memoria, una magnífica resolución, 16 colores, efectos tridimensionales con sprites, un sonido equivalente al de un sintetizador, un teclado profesional con 62 caracteres gráficos, toda una amplia gama de periféricos, la más completa gama de programas educativos, profesionales y de videojuegos...; en resumen, cuando se es un ordenador personal como no existe ningún otro en el mercado y el más vendido mundialmente, es muy difícil decir sin orgullo que eres un Commodore-64.

Claro que más difícil todavía es decir sin orgullo que tienes un Commodore-64. ¿Por qué no lo comprueba?

La revisión 03 del Kernal del C-64

Commodore 64: Nueva ROM de Kernal (Versión 03)

Por Jordi SASTRE

Normalmente en todos los ordenadores se van corrigiendo (se revisan) los programas que se incluyen en el equipo durante el proceso de fabricación, una de las ventajas del C-64 es la de que no es necesario cambiar el circuito integrado que contiene el sistema operativo cada vez que aparece una versión nueva, simplemente se copia en Ram y se cambia lo que haga falta...

C-64

Recientemente ha aparecido una nueva ROM de Kernal en los C-64. Esta ROM, identifican como versión -03, incorpora ciertos cambios en el sistema operativo, incluyendo el poder leer en el C-64 cassettes grabados por el VIC-20. Es posible adaptar esta nueva versión a los aparatos equipados con la ROM anterior (versión -02) mediante un programa que vuelque el Kernal a RAM, efectúe las modificaciones, y deje al ordenador trabajando con el Kernal modificado en RAM.

Los cambios realizados en la nueva ROM son:

1) Cuando se efectúa un POKE a la pantalla, aparece el carácter con el color actualmente en curso, sin necesidad de pokear la memoria de color. En la versión anterior (-02) al hacer esto el carácter quedaba "invisible" a menos que se pusiera el cursor encima o se le asignara un color mediante el POKE correspondiente. Esto permite identificar rápidamente la versión de ROM de la que está dotado un determinado aparato: se borra la pantalla, se desplaza el cursor un par de líneas hacia abajo y se ejecuta: POKE 1024,1. Si en el vértice superior izquierdo de la pantalla aparece la letra "A" significa que se trata de la nueva versión -03, en caso contrario, si no aparece nada, se trata de la -0,2.

2) Está corregido el bug de edición en la línea 25 (última de la pantalla), que se hacía notar al borrar caracteres (DEL) en esta línea.

3) También corregido el bug del INPUT que incluía como respuesta el texto que se entregaba como pregunta cuando la longitud de esta respuesta obligaba a doblar la línea.

4) Las rutinas que manejan el RS-232 no gestionan bien el flag de recepción del primer bit del carácter.

A continuación facilitamos un programa "modificador" de la ROM. De esta manera, quien posea un C-64 cuya ROM de Kernal sea de la versión -02 (hacer la prueba del POKE 1024,1) puede fácilmente actualizarse a la nueva versión -03. Este programa carga, en primer lugar, una rutina en código máquina para copiar el contenido actual de las ROMs sobre RAM (líneas 100 a 130). Esta rutina, que entra en acción con la línea 140, copia el contenido de las posiciones comprendidas entre 40960 y 49151, y entre 57344 y 65535. Esta operación se realiza en código máquina porque, de hacerla en Basic, puede resultar bastante lenta (16384 bytes a copiar!). Las líneas 210 y 220 efectúan los cam-

bios necesarios en la ROM (ahora sobre RAM), indicados en las líneas de DATAs a partir de la 500. Finalmente, la línea 310 activa la RAM para que entren en funcionamiento las modificaciones efectuadas.

```
100 REM *** COPIA DE ROM A
    RAM ***
110 READ D : D1=D : CH=0
120 READ X : IF X 256 THEN
    POKE D,X : CH=CH+X :
    D=D+1 : GOTO 120
130 IF X CH THEN PRINT "ERROR
    DE CHECKSUM" : STOP
140 SYS D1
150 :
200 REM *** MODIFICA BYTES ***
210 READ X : IF X 255 THEN
    D=X : GOTO 210
220 IF X 999 THEN POKE D,X :
    D=D+1 : GOTO 210
230 :
300 REM *** ACTIVA RAM ***
310 POKE 1,53
320 END
330 :
400 REM *** RUTINA DE
    COPIA ***
410 DATA 828
420 DATA 160,0,132,2,169,160,133,3,
    177,2,145,2,200,208,249
430 DATA 230,3,165,3,208,1,96,201,
    192,208,238,169,224,208,232
440 DATA 4120
450 :
500 REM *** BYTES A
    CAMBIAR ***
510 DATA 58540,129
520 DATA 58579,133,169,169,1,133,
    171,96,173,134,2
530 DATA 58748,32,240,233,169,39,
    232,180,217,48,6,24,105,40,232,16
540 DATA 58763,246,133,213,76,36,
    234,228,201,240,3,76,237,230,96,
    234
550 DATA 58914,145,229
560 DATA 59911,32,218,228,169,32,
    145,209,136,16,246,96,234
570 DATA 61332,76,211,228
580 DATA 65408,3
590 DATA 999
```

Para los curiosos facilito a continuación la rutina de codigo máquina listada en assembler:

```
000                        *=$033C      ;DIRECCION DE INICIO: 828
033C   A0 00     START     LDY #$00     ;INICIALIZA PUNTERO
033E   84 02               STY $02
0340   A9 A0               LDA #$A0
0342   85 03     BUCLE     STA $03
0344   B1 02     LEE       LDA ($02),   Y;COPIA BYTE
0346   91 02               STA ($02),Y
0348   C8                  INY          ;BUCLE DE 256 BYTES
0349   D0 F9               BNE LEE
034B   E6 03               INC $03      ;INCREMENTA PUNTERO
                                        CADA 256
034D   A5 03               LDA $03
034F   D0 01               BNE TEST     ;HA LLEGADO A $FFFF?
0351   60                  RTS          ;FIN DE COPIA
0352   C9 C0     TEST      CMP #$C0
0354   D0 EE               BNE LEE#;SI LLEGA A $C000,...
0356   A9 E0               LDA #$E0     ;...SIGUE EN $E000
0358   D0 E8               BNE BUCLE;Y SIGUE COPIANDO
```

club commodore

Compartiendo Experiencias entre amigos

Esta sección está dedicada a la colaboración de todos nuestros lectores y está dividida en dos partes:
1) Programación: Programas y similares
2) Magia
3) Trucos, sugerencias, etc.
habrá premios y alicientes "para todos los participantes".
*Todas las colaboraciones deben venir escritas a máquina a doble espacio y los programas grabados en cinta (si es posible), o con el listado completo en impresora. Nuestros lectores más jóvenes pueden escribir a mano **pero con letra muy clara.***
Enviarnos vuestra dirección para que podáis poneros en contacto unos con otros. ¡Animaros, chicos (...y chicas)!



Clave para interpretar los listados

Todos los listados que se publican en esta Revista han sido ejecutados en el modelo correspondiente de la gama de ordenadores COMMODORE.

Para los programas destinados a los ordenadores VIC-20 y COMMODORE 64, en los que se usan frecuentemente las posibilidades gráficas del teclado, se han sustituido los símbolos gráficos que aparecen normalmente en los listados por una serie de letras entre corchetes [] que indican la secuencia de teclas que se deben pulsar para obtener el carácter deseado. A continuación se da una tabla para aclarar la interpretación de las indicaciones entre corchetes:

[CRSRD] = Tecla cursor hacia abajo (sin SHIFT)
[CRSRU] = Tecla cursor hacia arriba (con SHIFT)
[CRSRR] = Tecla cursor a la derecha (sin SHIFT)
[CRSRL] = Tecla cursor a la izquierda (con SHIFT)
[HOME] = Tecla CLR/HOME (sin SHIFT)
[CLR] = Tecla CLR/HOME (con SHIFT)

Las indicaciones [BLK] a [YEL] corresponden a la pulsación de las teclas de 1 a 8 junto a la tecla CTRL. Lo mismo sucede con [CVSON] y [RVSOF] respecto a la tecla CTRL y las teclas 9 y 10.

El resto de las indicaciones constan de la parte COMM o SHIF seguidas de una letra, número o símbolo —por ejemplo [COMM+] o [SHIFA]—. Esto indica que para obtener el gráfico necesario en el programa deben pulsarse simultáneamente las teclas COMMODORE (la que lleva el logotipo) o una de SHIFT y la tecla indicada por la letra, el número o el símbolo, en el ejemplo anterior: COMMODORE y + o SHIFT y A, respectivamente.

En los signos gráficos además se cuenta el número de veces que aparece. Por ejemplo, [7 CRSRR] equivale a 7 pulsaciones de la tecla cursor a la derecha y [3 SPC] tres pulsaciones de la barra espaciadora. ■

MAT + DET

C-64

Me llamo Luis. Tengo 21 años. Soy estudiante y tengo un Commodore-64 desde noviembre del 83.
Me interesa la programación como tal, y los programas de carácter científico-técnico y aplicaciones.
Como no puedo permitirme los lujos de comprar Software, todos los programas que tengo son originales, copiados o fusilados.

En esta línea os envío este programa grabado en la cinta que os adjunto como "MAT & DET". Es un programa escrito totalmente en BASIC que multiplica dos matrices de cualquier orden o calcula un determinante.

Puesto que el C-64 no tiene problemas de memoria, el programa indica en cada momento al usuario las operaciones que este debe realizar, permitiendo incluso que se equivoque, dándole una nueva oportunidad en la introducción de datos.

Se puede calificar "lento" a este programa, pero para quien esté habituado a hacer ambas operaciones a mano o con calculadora, resolver un determinante de 7×7 en 4 segundos o multiplicar dos matrices de 10×10 en 29 seg., le parecerá una velocidad vertiginosa.

Como curiosidad al final de cada operación indica el tiempo invertido.

Aprovecho la oportunidad para manifestar mi voluntad de ponerme en contacto con programadores, aficionados a este tipo de programas científico-técnicos.

A continuación os doy una descripción del programa:

50000-50190	PREGUNTA POR LA OPCION DESEADA
50200-50340	INTRODUCE ORDEN DE LAS MATRICES A MULTIPLICAR
50350-50430	INTRODUCE PRIMERA MATRIZ
50440-50500	INTRODUCE SEGUNDA MATRIZ
50510-50630	MULTIPLICA LAS MATRICES Y ESCRIBE RESULTADO
50640-50650	ESPERA UNA TECLA
50660	VUELVE A COMENZAR
50700-50880	INTRODUCE ORDEN Y ELEMENTOS DEL DETERMINANTE
50890-51030	CALCULA Y ESCRIBE SOLUCION
51040-51050	ESPERA UNA TECLA
51060	VUELVE A COMENZAR
51100-51140	INTERCAMBIA DOS FILAS

El programa calcula un determinante triangularizando la matriz correspondiente. Es decir: si saliendo del programa deseamos ver la matriz de la cual hemos calculado el determinante, y escribimos (modo directo):
FOR I=1TO N: FOR J=1 TO N: ? A (I,J); : NEXT : ? : NEXT.
no obtendremos la matriz original introducida, sino su correspondiente triangularizada.

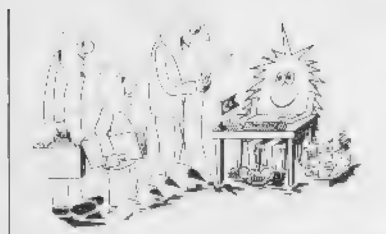
La numeración alta de las líneas del programa permite introducirlo como subrutina de programas más completos (...ojo con los CLR y con el redimensionamiento).

Luis Pellicer Lorenzo
c/Conde de Torrecedeira, 122, 3º A.
VIGO-8 (Pontevedra).


```
50000 REM##############################
###
50010 REM##
 ##
50020 REM##"[2SPC][SHIFTP][SHIFTR][SHIFTO]
[SHIFTD][SHIFTU][SHIFTC][SHIFTT][SHIFTO]
[SPC][SHIFTD][SHIFTE][SPC][SHIFTM][SHIFTA]
[SHIFTT][SHIFTR][SHIFTI][SHIFTC][SHIFTE]
[SHIFTS][2SPC]"##
50030 REM##"[10SPC]&[13SPC]"##
50040 REM##"[2SPC][SHIFTC][SHIFTA][SHIFTL]
[SHIFTC][SHIFTU][SHIFTL][SHIFTO][SPC]
[SHIFTD][SHIFTE][SHIFTT][SHIFTE][SHIFTR]
[SHIFTM][SHIFTI][SHIFTN][SHIFTA][SHIFTN]
[SHIFTT][SHIFTE][2SPC]"##
50050 REM##"[24COMMT]"##
50060 REM##"[2SPC]AUTOR:[SPC][SHIFTL]
[SHIFTU][SHIFTI][SHIFTS][SPC][SHIFTP]
[SHIFTE][2SHIFTL][SHIFTI][SHIFTC][SHIFTE]
[SHIFTR][2SPC]"##
50070 REM##
 ##
50080 REM##############################
###
50100 PRINT"[CLR]"
50110 POKE53272,23:PRINTCHR$(8)
50120 PRINT"[2CRSRD][7CRSRR][BLK]***
********************[ERROR]"
50130 PRINT"[7CRSRR][BLK]***[SPC][WHT]
[SPC][SHIFTT][SHIFTE][SHIFTC][SHIFTL]
[SHIFTE][SHIFTA][SPC][SHIFTO][SHIFTP]
[SHIFTC][SHIFTI][SHIFTO][SHIFTN][2SPC]
[BLK]***[ERROR]"
50140 PRINT"[7CRSRR][BLK]***********
***********[ERROR]"
50150 PRINT"[CRSRD][2CRSRR][CYN]1)[SPC]
[WHT][SHIFTP]RODUCTO[SPC][SHIFTM]ATR
ICES[SPC]"
50160 PRINT"[CRSRD][2CRSRR][CYN]2)[SPC]
[WHT][SHIFTC]ALCULO[SPC][SHIFTD]ETER
MINANTE[ERROR]"
50170 GETA$:IFA$=""THEN50170
50180 IFA$<>"1"ANDA$<>"2"THEN50170
50190 IFA$="2"THEN50700
50200 CLR
50210 REM****************************
***
50220 REM**
 **
50230 REM**  PRODUCTO DE MATRICES
 **
50240 REM**
 **
50250 REM****************************
```



```
***
50260 PRINT"[CLR][2CRSRD][4CRSRR][BLK]
**[2CRSRR][WHT][SHIFTP][SHIFTR][SHIFTO]
[SHIFTD][SHIFTU][SHIFTC][SHIFTT][SHIFTO]
[SPC][SHIFTD][SHIFTE][SPC][SHIFTM][SHIFTA]
[SHIFTT][SHIFTR][SHIFTI][SHIFTC][SHIFTE]
[SHIFTS][2CRSRR][BLK]**[ERROR]"
50270 PRINT"[CYN][CRSRD][2CRSRR][SHIFTI]
NTRODUCE[SPC]ORDEN[SPC][SHIFTM]ATRIC
ES[SPC]:"
50280 PRINT"[2CRSRD][CRSRR][CYN][SHIFTO]
RDEN[SPC][SHIFTM]ATRIZ[SPC][SHIFTA][SPC]
:"
50290 INPUT"[CRSRD][3CRSRR][CYN][SHIFTF]
ILAS[SPC][WHT]";N:IFN=0THENPRINT"[3CRSRU]
":GOTO50290
50300 INPUT"[CRSRD][3CRSRR][CYN][SHIFTC]
OLUMNAS[SPC][WHT]";M:IFM=0THENPRINT"
[3CRSRU]":GOTO50300
50310 PRINT"[2CRSRD][CRSRR][CYN][SHIFTO]
RDEN[SPC][SHIFTM]ATRIZ[SPC][SHIFTB][SPC]
:"
50320 PRINT"[CRSRD][3CRSRR][CYN][SHIFTF]
ILAS[SPC][WHT]?";M
50330 INPUT"[CRSRD][3CRSRR][CYN][SHIFTC]
OLUMNAS[SPC][WHT]";P:IFP=0THENPRINT"
[3CRSRU]":GOTO50330
50340 DIMA(N,M),B(M,P),C(N,P)
50350 PRINT"[2CRSRD][2CRSRR][CYN][SPC]
[SHIFTI]NTRODUCE[SPC]ELEMENTOS[SPC]:
[2CRSRD]"
50360 FORI=1TON:FORJ=1TOM
50370 PRINT"[WHT][CRSRU][2CRSRR][SHIFTA]
("I","J")[SPC]=[SPC]";:INPUTA(I,J)
50380 PRINT"[CRSRU][29SPC]"
50390 NEXTJ,I
50400 PRINT"[CYN][CRSRU][2CRSRR][SHIFTS]
ON[SPC]CORRECTOS[SPC]([SPC][SHIFTS][SPC]
/[SPC][SHIFTN][SPC])[SPC]?"
50410 GETA$:IFA$<>"S"ANDA$<>"N"THEN5
0410
50420 IFA$="N"THENPRINT"[CRSRU][34SPC]
":GOTO50360
50430 PRINT"[CRSRU][29SPC]"
50440 FORI=1TOM:FORJ=1TOP
50450 PRINT"[WHT][CRSRU][2CRSRR][SHIFTB]
("I","J")[SPC]=[SPC]";:INPUTB(I,J)
50460 PRINT"[CRSRU][29SPC]"
50470 NEXTJ,I
50480 PRINT"[CYN][CRSRU][2CRSRR][SHIFTS]
ON[SPC]CORRECTOS[SPC]([SPC][SHIFTS][SPC]
/[SPC][SHIFTN][SPC])[SPC]?"
50490 GETA$:IFA$<>"S"ANDA$<>"N"THEN5
0490
50500 IFA$="N"THENPRINT"[CRSRU][36SPC]
":GOTO50440
50510 PRINT"[CLR]"
50520 R1=TI:FORI=1TON
50530 IFI=N/2ORI=N/2+.5THENPRINTTAB(
2)"[WHT][SPC][SHIFTC][SPC]=[SPC][CYN]
";
50540 PRINTTAB(7)"[WHT][COMMH][CYN]"
;
50550 FORJ=1TOP
50560 FORS=1TOM
50570 C(I,J)=A(I,S)*B(S,J)+C(I,J)
50580 NEXTS
50590 C(I,J)=INT(C(I,J)*1E4)/1E4
50600 PRINTC(I,J);
50610 NEXTJ
50620 PRINT"[WHT][COMMN][CYN]"
50630 NEXTI:R2=TI:PRINT"[3CRSRD][2CRSRR]
[WHT][SHIFTT]IEMPO[SPC][SHIFTE]MPLEA
DO[SPC]:[SPC]"(R2-R1)/60"SEG."
50640 PRINT"[2CRSRD][19CRSRR][WHT][SHIFTP]
[SHIFTU][SHIFTL][SHIFTS][SHIFTA][SPC]
[SHIFTU][SHIFTN][SHIFTA][SPC][SHIFTT]
[SHIFTE][SHIFTC][SHIFTL][SHIFTA]";
50650 GETA$:IFA$=""THEN50650
50660 RUN
50700 REM***************************
***
50710 REM**
 **
50720 REM**  CALCULO DETERMINANTE
 **
50730 REM**
 **
50740 REM***************************
***
50750 CLR
50760 PRINT"[CLR][2CRSRD][4CRSRR][BLK]
**[2CRSRR][WHT][SHIFTC][SHIFTA][SHIFTL]
[SHIFTC][SHIFTU][SHIFTL][SHIFTO][SPC]
[SHIFTD][SHIFTE][SHIFTT][SHIFTE][SHIFTR]
[SHIFTM][SHIFTI][SHIFTN][SHIFTA][SHIFTN]
[SHIFTT][SHIFTE][BLK][2SPC]**[ERROR]
[CRSRD]"
50770 INPUT"[CYN][2CRSRR][SHIFTI]NTR
ODUCE[SPC]ORDEN[SPC][SHIFTD]ETERMINA
NTE[SPC]:[SPC]";N
50780 IFN=0THENPRINT"[2CRSRU][33SPC]
":GOTO50770
50790 DIMA(N,N),S(N)
50800 PRINT"[2CRSRD][3CRSRR][CYN][SHIFTI]
NTRODUCE[SPC]ELEMENTOS[SPC]:[2CRSRD]
"
50810 FORI=1TON:FORJ=1TON
50820 PRINT"[WHT][CRSRU][2CRSRR][SHIFTA]
("I","J")[SPC]=[SPC]";:INPUTA(I,J)
50830 PRINT"[CRSRU][29SPC]"
50840 NEXTJ,I
50850 PRINT"[CYN][CRSRU][2CRSRR][SHIFTS]
ON[SPC]CORRECTOS[SPC]([SPC][SHIFTS][SPC]
/[SPC][SHIFTN][SPC])[SPC]?"
50860 GETA$:IFA$<>"S"ANDA$<>"N"THEN5
0860
50870 IFA$="N"THENPRINT"[CRSRU][34SPC]
":GOTO50810
50880 PRINT"[CRSRU][34SPC]"
50890 R1=TI
50900 FORI=1TON-1
50910 FORT=I+1TON
50920 B=A(I,T):C=A(I,I)
50930 IFB=0ANDC=0THEN50980
50940 IFC=0THENGOSUB51100:GOTO50900
50950 FORS=ITON
50960 A(S,T)=A(S,T)-(B/C)*A(S,I)
50970 NEXTS
50980 NEXTT,I
50990 L=1:FORR=1TON
51000 L=A(R,R)*L
51010 NEXTR
51020 R2=TI:PRINT"[2CRSRD][3CRSRR][WHT]
[SHIFTS]OLUCION[SPC]:[SPC]"((-1)↑D)*
L
51030 PRINT"[2CRSRD][ERROR][3CRSRR][SHIFTT]
IEMPO[SPC]EMPLEADO[SPC]:"(R2-R1)/60"
SEG."
51040 PRINT"[2CRSRD][18CRSRR][SPC][WHT]
[SHIFTP][SHIFTU][SHIFTL][SHIFTS][SHIFTA]
[SPC][SHIFTU][SHIFTN][SHIFTA][SPC][SHIFTT]
[SHIFTE][SHIFTC][SHIFTL][SHIFTA]";
51050 GETA$:IFA$=""THEN51050
51060 RUN
51100 D=D+1
51110 FORK=1TON
51120 S(K)=A(K,T-1):A(K,T-1)=A(K,T)
51130 A(K,T)=S(K)
51140 NEXT:RETURN
```



DADOS



C-64

Os mando este programa que aunque puede parecer un poco complicado es muy sencillo de funcionamiento.

Se trata de un juego de dados con figuras y la forma de juego es como la del POKER.

El ordenador "pasa" una jugada y hay que superarla, si no se consigue, se anotará un punto.

Sólo se puede tirar una vez cada dado, en cada jugada.

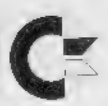
Para empezar de nuevo hay que pulsar "RETURN" y para terminar la jugada se debe pulsar "F1".

Si os animáis a probarlo, espero que os divirtáis y que no os gane siempre, un saludo y hasta pronto.

Luis Pons Minguell
Rambla Nova, 123
Tarragona. - Telf.: 23 89 24.

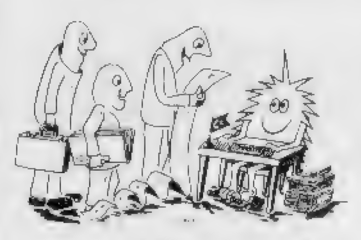
```
5 REM ------  LUIS PONS MINGUELL ---
--
6 REM ----------- DADOS ----------
--
10 PRINT"[CLR]"
20 POKE53280,0:POKE53281,0
30 CLR:J=1:R$="[27SPC]"
40 PRINTCHR$(31)"[4CRSRD][RVSON]",R$

50 PRINTCHR$(30)"[RVSON]";R$
60 PRINTCHR$(5)"[RVSON][SPC]PARA[SPC]
EMPEZAR[SPC]PULSE[SPC]RETURN[SPC]"
65 PRINTCHR$(30)"[RVSON]";R$
68 PRINTCHR$(31)"[RVSON]";R$
70 GET W$:IF W$<>CHR$(13) GOTO 70
80 PRINT"[CLR]":PRINTCHR$(5):B(1)=0:
J=1
90 FOR N=1 TO 5
100 V(N)=INT(6*RND(1))+1
102 VS(N)=0
110 NEXT N
115 IF V(1)=5 OR V(2)=6 THEN V(1)=V(
2)
118 V(5)=V(1)
120 FOR N=1 TO 5
130 V$(N)="[SPC]":W$(N)="[SPC]":Y(N)
=0:B(2)=0:VA(N)=0
135 NEXTN
150 REM
160 FOR  N=1 TO 5
170 IF V(N)=1 THEN V$(N)="[SPC]DE[SPC]
ASES"
171 IF V(N)=2 THEN V$(N)="[SPC]DE[SPC]
REYES"
172 IF V(N)=6 THEN V$(N)="[SPC]DE[SPC]
NEGROS"
173 IF V(N)=3 THEN V$(N)="[SPC]DE[SPC]
DAMAS"
175 IF V(N)=4 THEN V$(N)="[SPC]DE[SPC]
JOTAS"
178 IF V(N)=5 THEN V$(N)="[SPC]DE[SPC]
ROJOS"
275 IF V(1)=V(N) THEN  Y(1)=Y(1)+1
295 IF V(2)=V(N) THEN  Y(2)=Y(2)+1
305 IF V(3)=V(N) THEN  Y(3)=Y(3)+1
315 IF V(4)=V(N) THEN  Y(4)=Y(4)+1
325 IF V(5)=V(N) THEN  Y(5)=Y(5)+1
328 NEXT N
330 FOR N=1 TO 5
350 IFY(N)=1THENW$(N)="UNO":B(J)=B(J
)+1
351 IFY(N)=2THENW$(N)="PAREJA" B(J)=
B(J)+8
353 IFY(N)=3THENW$(N)="TRIO":B(J)=B(
J)+16
355 IFY(N)=4THENW$(N)="POKER" B(J)=B
(J)+32
358 IFY(N)=5THENW$(N)="REPOKER" BN=B
(J)+64
400 NEXT N
1000 A$=W$(1)+V$(1)
1010 C$=W$(3)+V$(3)
1012 B$=W$(2)+V$(2)
1015 G$=W$(4)+V$(4)
1020 F$=W$(5)+V$(5)
1110 IF A$=B$ THEN B$=""
1120 IF A$=C$ THEN C$=""
1130 IF A$=G$ THEN G$=""
1140 IF A$=F$ THEN F$=""
```



```
1150 IF B$=C$ THEN C$=""
1160 IF B$=G$ THEN G$=""
1170 IF B$=F$ THEN F$=""
1180 IF C$=G$ THEN G$=""
1190 IF C$=F$ THEN F$=""
1200 IF G$=F$ THEN F$=""
5000 PRINT"[CLR]":PRINTCHR$(30)
5010 PRINTTAB(7)"[SHIFTU][3SHIFTC][SHIFTI]
[3SPC][SHIFTU][3SHIFTC][SHIFTI][2SPC]
[SHIFTU][3SHIFTC][SHIFTI]"
5035 PRINTTAB(7)"[SHIFTB][3SPC][SHIFTB]
[3SPC][SHIFTB][3SPC][SHIFTB][2SPC][SHIFTB]
[3SPC][SHIFTB]"
5040 PRINTTAB(7)"[SHIFTJ][3SHIFTC][SHIFTK]
[3SPC][SHIFTJ][3SHIFTC][SHIFTK][2SPC]
[SHIFTJ][3SHIFTC][SHIFTK]"
5050 PRINTTAB(11)"[SHIFTU][3SHIFTC][SHIFTI]
[3SPC][SHIFTU][3SHIFTC][SHIFTI]"
5070 PRINTTAB(11)"[SHIFTB][3SPC][SHIFTB]
[3SPC][SHIFTB][3SPC][SHIFTB]"
5080 PRINTTAB(11)"[SHIFTJ][3SHIFTC][SHIFTK]
[3SPC][SHIFTJ][3SHIFTC][SHIFTK]"
5090 FOR T=1 TO 2
5100 PRINT"[3SPC][SHIFTU][26SHIFTC][SHIFTI]
"
5110 PRINT"[3SPC][SHIFTB][26SPC][SHIFTB]
"
5120 PRINT"[3SPC][SHIFTB][26SPC][SHIFTB]
"
5130 PRINT"[3SPC][SHIFTB][26SPC][SHIFTB]
"
5140 PRINT"[3SPC][SHIFTB][26SPC][SHIFTB]
"
5150 PRINT"[3SPC][SHIFTB][26SPC][SHIFTB]
"
5170 PRINT"[3SPC][SHIFTJ][26SHIFTC][SHIFTK]
"
5180 NEXT T
5181 PRINT"[5CRSRU]"
5182 PRINTTAB(4)"[SHIFTU][5SHIFTC][SHIFTI]
"TAB(22)"[SHIFTU][5SHIFTC][SHIFTI]"
5184 PRINTTAB(4)"[SHIFTB]TU[3SPC][SHIFTB]
"TAB(22)"[SHIFTB]YO[3SPC][SHIFTB]"
```

```
5185 PRINTTAB(4)"[SHIFTJ][5SHIFTC][SHIFTK]
"TAB(22)"[SHIFTJ][5SHIFTC][SHIFTK]"
5200 FOR H=1 TO 5
5210 IF V(H)=1 THEN I$(H)="[SHIFTQ]"
5220 IF V(H)=2 THEN I$(H)="K"
5230 IF V(H)=3 THEN I$(H)="Q"
5240 IF V(H)=4 THEN I$(H)="J"
5250 IF V(H)=5 THEN I$(H)="R"
5260 IF V(H)=6 THEN I$(H)="N"
5270 NEXT H
5300 PRINTCHR$(5)"[HOM][2CRSRD]"
5310 PRINTTAB(9)I$(1);
5320 PRINTTAB(17)I$(2);
5330 PRINTTAB(24)I$(3)
5340 PRINT"[2CRSRD]"TAB(13)I$(4);
5350 PRINTTAB(21)I$(5)
5370 PRINT"[2CRSRD]"TAB(10)A$
5380 PRINTTAB(10)B$
5390 PRINTTAB(10)C$
5395 PRINTTAB(10)G$
5410 PRINTTAB(10)F$
5500 PRINTTAB(6)"[2CRSRD]QUE[SPC]DAD
O[SPC]QUIERES[SPC]TIRAR?"
5505 J=2
5510 PRINTTAB(12)"1[3SPC]2[3SPC]3"
5520 PRINTTAB(14)"4[3SPC]5"
5530 PRINTTAB(7);TUTAB(25)YO
5535 PRINTTAB(12)"JUGADA=F1";
5540 GET EE$
5550 EE=INT(VAL(EE$))
5555 IF EE>0 AND EE<6 THEN 6000
5560 IF EE$="[F1]" GOTO 7000
5565 IF EE$=CHR$(13)GOTO 10
5570 GOTO 5540
6000 VS(EE)=VS(EE)+1
6010 IF VS(EE)>1 GOTO 5540
6020 V(EE)=INT(6*RND(1))+1
6030 GOTO 120
7000 IF B(2)<=B(1) THEN YO=YO+1
7010 IF B(2)>B(1) THEN TU=TU+1
7030 GOTO 80
```



PROGRAMA ARACNIDA

C-64

Antes que nada os quiero felicitar por vuestra revista porque me ha ayudado mucho y me ha enseñado muchas cosas que no sabía aunque hay cosas que no entiendo todavía.

Os presento mi programa "**ARACNIDA**" para **Commodore 64**, por si os puede interesar.

Todo empezó con unas simples pruebas con los sprites y tras 19 mejoras importantes (porque las no importantes no las guardo) he conseguido este programa (cosa que no esperaba al principio).

Empieza la batalla

Nada más empezar el programa el ordenador te preguntará la velocidad de la nave y de la bala. El máximo es 9 y el mínimo 0, pero si pones 0 no te puedes mover.

Luego empieza la batalla, tú manejas la nave que aparece abajo.

Los mandos son:

@ Arriba
/Abajo
:Izquierda
;Derecha
(Espacio) Disparos

Tienes que matar las seis arañas pero sin que ellas te coman. Si matas a todas las arañas el ordenador te dirá el tiempo y si has batido el récord.

Si te matan tres veces simplemente te preguntará a ver si quieres volver a empezar.

Los rombos que aparecen arriba a la izquierda son las naves que te quedan.

Quiero explicaros brevemente el programa:

Lineas	Función
10	Activa las teclas repetitivas
15-50	Te pregunta la rapidez de la bala y de la nave
60-95	Poner valores iniciales a los Datos



100	Dibuja los rombos
120-140	Activa los sprites
150-170	Lee los datas
180-220	Determina y dibuja los sprites
240-310	Recoge con GETS el movimiento
320-370	Si te matan...
380-430	Si matas...
440-660	Pone en marcha los rebotes
670-720	Movimientos
2000-4007	Datas
6000-6020	SONIDO
10000-20030	Si ganas
20210-20340	Si pierdes

¡Ah! Y gracias a José Ramón Lasa por su programa de Editor de sprites porque me ha ahorrado mucho tiempo y trabajo para crear mis arañas.

Carlos Beldarrain Santos
11 años de edad
(¡¡¡si, sólo 11, estoy en 6º de E.G.B.!!!)
C/Hermanos Imaz, 1-7º D.
PAMPLONA. Teléf.: 23 79 92

```
0 REM **************
1 REM *            *
2 REM *  ARACNIDA  *
3 REM *    POR:    *
4 REM *            *
5 REM *   CARLOS   *
6 REM * BELDARRAIN *
7 REM *            *
8 REM **************
10 POKE650,255
15 L=127:PRINT"[CLR][CRSRD][3SPC]RAP
IDEZ[SPC]DE[SPC]LA[SPC]NAVE[SPC]1-9[SPC]
"
20 GETLY$:IFLY$=""THEN20
25 LY=VAL(LY$)
30 LR=LY:IFT=0THENT=1000
35 PRINT"[2CRSRD][3SPC]RAPIDEZ[SPC]D
E[SPC]LA[SPC]BALA[SPC]1-9[WHT][SPC]"

40 GET MN$:IFMN$=""THEN40
50 MN=VAL(MN$)
60 PRINT"[ERROR]":TI$="000000"
70 PRINT"[CLR]":ASD=INT(RND(5)*200)+
20
80 IFASD/2<>INT(ASD/2)THENASD=ASD+1
90 RE=0:MR=0:NA=3
93 F=ASD:DF=2:G=ASD:DG=2:C=172:H=10:
DH=5:I=28:DI=4:J=50:DJ=4:K=L+5:M=210
:MY=210
95 V=53248:X=70:DX=8:Y=70:DY=10:A=84
:DA=2:B=76:DB=2:D=150:DD=2:E=150:DE=
2:NN=0
100 POKE53281,8
110 FORLA=1TO3:POKE1024+LA,90:NEXTLA

120 POKEV+21,255
130 POKE2042,192:POKE2043,192:POKE20
44,192:POKE2040,192:POKE2041,192
140 POKE2046,198:POKE2047,199:POKE20
45,192
145 RESTORE
150 FORN=0TO62:READQ:POKE12288+N,Q:N
EXT
160 FORN=0TO62:READZ:POKE12736+N,Z:N
EXT
170 FORN=0TO62:READZ:POKE12672+N,Z:N
EXT
180 POKEV+23,78:POKEV+29,78:POKEV+39
,15:POKEV+40,7:POKEV+41,15:POKEV+42,
15
190 POKEV+43,15:POKEV+44,15:POKEV+45
,2:POKEV+46,3
200 POKEV+4,X:POKEV+6,C:POKEV+8,D:PO
KEV+0,F:POKEV+2,H:POKEV+10,I:POKEV+1
2,L
210 POKEV+14,K:POKEV+15,M
220 POKE V+5,Y:POKEV+7,B:POKEV+9,E:P
OKEV+1,G:POKEV+3,ASD:POKEV+11,J:POKE
V+13,MY
230 IFRE=6THENGOTO10000
240 GETA$:IFA$=";"THENL=L+LR
250 IFA$=":"THENL=L-LR
260 IFA$="@"THENMY=MY-LR
270 IFA$="/"THENMY=MY+LR
290 GETB$:IFB$="[SPC]"THENNN=1
300 IFNN=1THENM=M-MN
310 IFM<30THENM=230:NN=0
320 IFMY<Y+10ANDMY>Y-10THENIFL<X+10A
NDL>X-10THEN20210
330 IFMY<B+10ANDMY>B-10THENIFL<C+10A
NDL>C-10THEN20210
340 IFMY<E+10ANDMY>E-10THENIFL<D+10A
NDL>D-10THEN20210
350 IFMY<G+10ANDMY>G-10THENIFL<F+10A
```





```
NDL>F-10THEN20210
360 IFMY<ASD+10ANDMY>ASD-10THENIFL<H
+10ANDL>H-10THEN20210
370 IFMY<J+10ANDMY>J-10THENIFL<I+10A
NDL>I-10THEN20210
380 IFM<Y+20ANDM>Y-10THENIFK<X+20AND
K>X-10ANDNN=1THENDX=0:Y=0:DY=0:M=28:
RE=RE+1
390 IFM<B+20ANDM>B-10THENIFK<C+20AND
K>C-10ANDNN=1THENDA=0:B=0:DB=0:M=28:
RE=RE+1
400 IFM<E+20ANDM>E-10THENIFK<D+20AND
K>D-10ANDNN=1THENDD=0:E=0:DE=0:M=28:
RE=RE+1
410 IFM<G+20ANDM>G-10THENIFK<F+20AND
K>F-10ANDNN=1THENDF=0:G=0:DG=0:M=28:
RE=RE+1
420 IFM<ASD+20ANDM>ASD-10THENIFK<H+2
0ANDK>H-10ANDNN=1THENDH=0:ASD=0:M=28
:RE=RE+1
430 IFM<J+20ANDM>J-10THENIFK<I+20AND
K>I-10ANDNN=1THENDJ=0:I=0:DI=0:M=28:
RE=RE+1
440 IFX>=254THENDX=-DX
450 IFX<22THENDX=-DX
460 IFY>230THENDY=-DY
470 IFY<50THENDY=-DY
480 IFA>234THENDA=-DA
490 IFA<=0THENDA=-DA
500 IFB>230THENDB=-DB
510 IFB<50THENDB=-DB
520 IFD>=254THENDD=-DD
530 IFD<20THENDD=-DD
540 IFE>230THENDE=-DE
550 IFE<50THENDE=-DE
560 IFF<18THENDF=-DF
570 IFF>250THENDF=-DF
580 IFG<18 THENDG=-DG
590 IFL>245THENL=245
600 IFMY>235THENMY=235
610 IFL<0THENL=0
620 IFMY<0THENMY=0
630 IFG>232THENDG=-DG
640 IFH<=0ORH>=255THENDH=-DH
650 IFI<20ORI>=252THENDI=-DI
660 IFJ<30ORJ>230THENDJ=-DJ
670 X=X+DX:Y=Y+DY:A=A+DA:B=B+DB:C=25
5-A:D=D+DD:E=E+DE:F=F+DF:G=G+DG:H=H+
DH
680 IFNN=0THENM=MY
690 IFNN=0THENK=L+5
700 I=I+DI:J=J+DJ
710 I=I+DI:J=J+DJ
720 GOTO200
2000 REM ARANIA
2001 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
2002 DATA 8, 0, 16, 28, 0, 56, 62, 0
, 124
2003 DATA 103, 60, 230, 195, 255, 19
5, 17, 255, 136
2004 DATA 59, 255, 220, 127, 126, 25
4, 207, 60, 243
2005 DATA 7, 153, 224, 23, 255, 232,
 63, 231, 252
2006 DATA 127, 129, 254, 96, 255, 6,
 224, 126, 7
2007 DATA 0, 0. 0, 0, 0, 0, 0, 0, 0
3000 REM NAVE
3100 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
3200 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
3300 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
3400 DATA 0, 0, 0, 0, 0, 0, 0, 16, 0
3500 DATA 0. 16, 0, 0, 16, 0, 0, 56,
 0
3600 DATA 0, 56, 0, 0, 124, 0, 0, 12
4, 0
3700 DATA 0, 238, 0, 0, 198, 0, 0, 1
98, 0
4000 REM OVNI
4001 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
4002 DATA 0, 0, 0, 0, 0, 0, 1, 224,
0
4003 DATA 15, 252, 0, 31, 254, 0, 30
, 30, 0
4004 DATA 30, 222, 0, 30, 30, 0, 15,
 252, 0
4005 DATA 1, 224, 0, 3, 48, 0, 6, 24
, 0
4006 DATA 12, 12, 0, 0, 0, 0, 0, 0,
0
4007 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
6000 POKE54277,130:POKE54273,2:POKE5
4272,140:POKE54278,130
6010 POKE54276,129:POKE54296,15
6012 FORID=1TO900:NEXTID
6015 POKE54276,15
6020 RETURN
10000 REM **** GANAS****
10005 POKE53281,7:POKEV+21,0
10010 PRINT"[WHT][10CRSRR][9CRSRD][RVSON]
HAS[3CRSRL][2CRSRD]VENCIDO[7CRSRL][2CRSRD]
AL[2CRSRL][2CRSRD]ENEMIGO[3CRSRD]";
10020 A=TI/60:PRINT"[CRSRD][7CRSRL][WHT]
TIEMPO:";A:PRINT"[6CRSRR][CRSRD]RECO
RD";T;"SEGUNDOS"
10030 PRINT"[GRN][HOM]"
10035 IF A<TTHENGOSUB20000
10040 PRINT"[WHT][5CRSRR]QUIERE[SPC]
VOLVER[SPC]A[SPC]EMPEZAR(S/N)"
10050 GETC$:IFC$=""THEN10050
10060 IFC$="S"THEN10
10065 IFC$<>"N"THEN10050
10070 PRINT"[WHT][5CRSRR][CRSRD]SE[SPC]
ACABO"
10080 POKE650,0:END
20000 PRINT"[4CRSRR][CRSRD][WHT]A[SPC]
CABAS[SPC]DE[SPC]BATIR[SPC]EL[SPC]RE
CORD"
20010 T=A:PRINT"NUEVO[SPC]RECORD:";T

20030 RETURN
20210 REM**** PIERDES ****
20215 NA=NA-1:MY=210:L=155:K=L+10:M=
MY:NN=0:GOSUB6000
20217 IFNA=1THENPOKE1024+2,32
20218 IFNA=2THENPOKE1024+3,32
20219 IFNA=0THENPOKE1024+1,32
20220 IFNA<>0THEN200
20225 POKE53281,7
20230 POKEV+21,0
20296 PRINT"[ERROR]"
20300 PRINT
20310 PRINT"[WHT][3CRSRD][2CRSRR]TE[SPC]
HA[SPC]VENCIDO"
20320 PRINT"[WHT][2CRSRD][5CRSRR]EL"

20330 PRINT"[WHT][2CRSRD][3CRSRR]ENE
MIGO[2CRSRD]"
20340 GOTO10040
```





MAQUINA DE FRUTAS

VIC-20

Queridos amigos del equipo Commodore World:
En primer lugar quiero felicitaros a todos los que hacéis la revista de Commodore World, ya que me gusta como habéis empezado así como también el contenido de la misma.

Yo por mi parte voy a intentar modestamente colaborar (si lo dais de paso) en esta ocasión con un juego que está muy de moda en bares y cafeterías como son las Máquinas de Frutas; supongo que hay otras versiones más o menos sencillas para los chavales como yo, pero he querido hacer la mía propia para practicar con el VIC-20 a la vez que me divierto. Paso a comentar brevemente las funciones de las líneas del programa.

De la 10-414	Generan las instrucciones data para los dibujos en alta resolución de las frutas. NOTA: si queréis crear otros dibujos, utilizar el editor de caracteres de la revista Club Commodore nº 3.
De 425-491	Contiene las instrucciones de las variables de los gráficos (Frutas).
De 492-499	Son las instrucciones que dibujan el marco que encierra a las frutas.
de 509-610	En estas líneas están las instrucciones que de forma aleatoria escoge una fruta, así como también contiene el sonido de la rotación.
de 617-663	En estas líneas están los premios obtenidos así como los avances, considerados como un posible premio.
de 665-692	Estas generan la música de los premios, que los hay menores, mayores y especial (JM×3).
de 1005-1150	Contienen las instrucciones para que de forma aleatoria imprima los avances.
de 1160-1170	Contiene las instrucciones Print que indican el seguir jugando o finalizar (por haber saltado la banca o haberse quedado sin blanca).

FORMA DE JUGAR: Después de cargar el programa se hace RUN y el programa entra en funcionamiento automáticamente y no se detiene hasta pulsar cualquier tecla; si se obtiene premio dará la puntuación. Si no sale ningún premio o para comenzar otra partida, pulsar una tecla cualquiera para poner en marcha y se deja rodar a voluntad hasta que se para con otra tecla; si sale un avance como posible premio, entonces pulsar la barra espaciadora (lo indica en pantalla). Este programa yo lo he metido en el VIC-20 con sólo ampliación de 3K, o sea que los que tengáis más Ks' podéis introducir más variantes en el juego, pero eso lo dejo a gusto del programador.

Espero que esta mi primera colaboración les sirva al equipo de Commodore World así como alguno de sus lectores.

Miguel Cortijo Maeso
13 años
C/Guatemala, 9 - 1º D. Valladolid-14
Teléf.: 23 08 76



```
5 REM***MAQUINA DE FRUTAS***
6 REM***POR MIGUEL A. CORTIJO***
7 REM*** 20-3-1984 ***
8 REM INSTRUCCIONES DATA
10 PRINT"[CLR]":POKE52,28:POKE56,28

20 POKE36869,255:POKE36879,158
25 FORI=0TO254
30 READA
32 POKEI+7168,A
35 NEXT
37 FORI=0TO7:READA:POKEI+7424,A:NEXT
:POKE36869,240
38 W=0:L=5:POKE36869,255
39 DATA0,255,255,255,255,255,255,0
40 DATA0,1,1,3,3,3,7,7
50 DATA224,0,0,128,128,128,192,192
60 DATA15,15,31,31,31,31,15,6
70 DATA224,224,240,240,240,240,224,1
92
80 DATA31,63,51,3,6,12,24,56
90 DATA128,0,128,192,96,96,112,216
100 DATA108,92,92,124,124,124,56,0
110 DATA184,184,248,248,248,112,0,0
120 DATA7,3,1,1,31,23,29,23
130 DATA0,0,0,96,240,80,240,80
140 DATA30,11,15,13,6,3,3,1
150 DATA240,224,160,224,192,128,128,
0
160 DATA1,2,1,7,15,15,15,8
170 DATA128,64,128,224,240,240,240,1
6
180 DATA23,15,56,38,92,60,30,7
190 DATA232,240,28,228,122,124,248,2
24
200 DATA3,7,7,15,15,31,31,31
210 DATA224,208,136,4,36,36,116,116
220 DATA31,31,31,15,15,7,7,3
230 DATA116,116,36,36,4,136,208,224
240 DATA7,9,0,0,2,7,9,9
250 DATA128,240,200,128,224,112,184,
248
360 DATA13,15,15,15,15,7,3,1
370 DATA248,248,248,248,248,240,224,
192
380 DATA3,57,29,31,27,25,25,25
390 DATA192,156,184,248,216,152,152,
152
400 DATA25,25,25,25,125,33,63,30
410 DATA152,152,152,152,190,128,0,0
411 DATA24,60,110,78,223,191,187,255

412 DATA253,255,223,251,126,110,60,2
4
413 DATA0,0,0,0,0,0,0,0
414 DATA0,0,0,0,0,0,0,0
420 REM VARIABLES FRUTAS
425 PRINT"[CLR]"
430 A$(1)="[CRSRD][2CRSRR][GRN]AB[CRSRD]
[2CRSRL]CD"
440 A$(2)="[CRSRD][2CRSRR][PUR]EF[CRSRD]
[2CRSRL]GH"
450 A$(3)="[CRSRD][2CRSRR][PUR]IJ[CRSRD]
[2CRSRL]KL"
460 A$(4)="[CRSRD][2CRSRR][CYN]MN[CRSRD]
[2CRSRL]OP"
470 A$(5)="[CRSRD][2CRSRR][GRN]QR[CRSRD]
[2CRSRL]ST"
480 A$(6)="[CRSRD][2CRSRR][RED]UV[CRSRD]
[2CRSRL]WX"
490 A$(7)="[CRSRD][2CRSRR][BLK]YZ[CRSRD]
[2CRSRL][£"
491 A$(8)="[CRSRD][2CRSRR][YEL][SPC]
][CRSRD][2CRSRL][SPC]↑"
```

```
492 PRINT"[CRSRU][PUR]@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@"
493 PRINT"[5CRSRD]@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@"
494 PRINT"[BLU][HOM][3CRSRD][2SHIFT@]
[CRSRD][2CRSRL][2SHIFT@][CRSRD][2CRSRL]
[2SHIFT@][CRSRD][2CRSRL][2SHIFT@][CRSRD]
[2CRSRL][2SHIFT@][CRSRD][2CRSRL][2SHIFT@]
"
495 PRINT"[HOM][3CRSRD][7CRSRR][2SHIFT@]
[CRSRD][2CRSRL][2SHIFT@][CRSRD][2CRSRL]
[2SHIFT@][CRSRD][2CRSRL][2SHIFT@][CRSRD]
[2CRSRL][2SHIFT@][CRSRD][2CRSRL][2SHIFT@]
"
496 PRINT"[HOM][3CRSRD][14CRSRR][2SHIFT@]
[CRSRD][2CRSRL][2SHIFT@][CRSRD][2CRSRL]
[2SHIFT@][CRSRD][2CRSRL][2SHIFT@][CRSRD]
[2CRSRL][2SHIFT@][CRSRD][2CRSRL][2SHIFT@]
"
497 PRINT"[HOM][3CRSRD][21CRSRR][2SHIFT@]
[CRSRD][2CRSRL][2SHIFT@][CRSRD][2CRSRL]
[2SHIFT@][CRSRD][2CRSRL][2SHIFT@][CRSRD]
[2CRSRL][2SHIFT@][CRSRD][2CRSRL][2SHIFT@]
"
498 PRINT"[HOM][3CRSRD][20CRSRR][2SHIFT@]
[CRSRD][2CRSRL][2SHIFT@][CRSRD][2CRSRL]
[2SHIFT@][CRSRD][2CRSRL][2SHIFT@][CRSRD]
[2CRSRL][2SHIFT@][CRSRD][2CRSRL][2SHIFT@]
[PUR]@"
499 PRINT"[HOM][PUR][12CRSRD][CRSRL]
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@"
505 REM MOVIMIENTO ALEATORIO DE LAS
FRUTAS
509 FORN=1TO3
510 A=INT(RND(1)*8)+1
520 A(N)=A
530 G=(N-1)*7+1
540 PRINTTAB(G)"[5CRSRD]"
550 PRINT"[HOM]"
560 PRINTTAB(G);A$(A)
570 POKE36878,15:POKE36876,200:FORZ=
1TO10:NEXT:POKE36876,0
571 PRINT"[HOM][2CRSRD]"
572 NEXTN
600 GETQ$:IFQ$<>""THEN617
610 GOTO509
615 REM LISTA DE PREMIOS
617 FORN=1TO3
625 IFA(1)=A(2)ANDA(2)=A(3)THENW=15:
GOTO665
627 IFA(1)=(7)AND(7)=A(2)ANDA(2)=(7)
AND(7)=A(3)THENW=20:GOTO665
629 IFA(2)=(1)AND(1)=A(3)THENGOTO101
0
630 IFA(2)=(2)AND(2)=A(3)THENGOTO101
0
631 IFA(2)=(3)AND(3)=A(3)THENGOTO101
0
632 IFA(2)=(4)AND(4)=A(3)THENGOTO101
0
633 IFA(1)=(7)AND(7)=A(3)THENW=4:GOT
O680
634 IFA(1)=(6)AND(6)=A(3)THENW=4:GOT
O680
635 IFA(1)=(5)AND(5)=A(3)THENW=4:GOT
O680
636 IFA(1)=(8)AND(8)=A(3)THENW=4:GOT
O680
637 IFA(1)=(5)AND(5)=A(2)THENW=7:GOT
O680
638 IFA(1)=(6)AND(6)=A(2)THENW=7:GOT
```

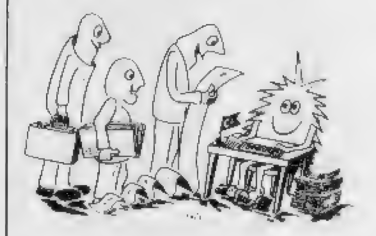


```
0680
639 IFA(1)=(7)AND(7)=A(2)THENW=7:GOT
0680
640 IFA(1)=(8)AND(8)=A(2)THENW=7:GOT
0680
641 IFA(1)=(1)AND(1)=A(3)THENW=4:GOT
0680
642 IFA(1)=(2)AND(2)=A(3)THENW=4:GOT
0680
643 IFA(1)=(4)AND(4)=A(3)THENW=4:GOT
0680
644 IFA(1)=(3)AND(3)=A(3)THENW=4:GOT
0680
645 IFA(1)=(7)AND(7)=A(3)ANDA(2)=(1)
THENW=15:GOTO665
646 IFA(1)=(1)AND(1)=A(3)ANDA(2)=(6)
THENW=15:GOTO665
647 IFA(1)=(2)AND(2)=A(3)ANDA(2)=(8)
THENW=15:GOTO665
648 IFA(1)=(4)AND(4)=A(3)ANDA(2)=(3)
THENW=15:GOTO665
649 IFA(1)=(5)AND(5)=A(3)ANDA(2)=(7)
THENW=15:GOTO665
650 IFA(1)=(1)AND(1)=A(2)THENW=7:GOT
0680
651 IFA(1)=(2)AND(2)=A(2)THENW=7:GOT
0680
652 IFA(1)=(3)AND(3)=A(2)THENW=7:GOT
0680
653 IFA(1)=(4)AND(4)=A(2)THENW=7:GOT
0680
660 IFA(2)=(5)AND(5)=A(3)THENGOTO101
0
661 IFA(2)=(6)AND(6)=A(3)THENGOTO101
0
662 IFA(2)=(7)AND(7)=A(3)THENGOTO101
0
663 IFA(2)=(8)AND(8)=A(3)THENGOTO101
0
664 GOTO1160:REM MUSICA
665 POKE36878,15:S2=36875:PRINT"[HOM]
[RVSON][BLK][15CRSRD]HAS[SPC]OPTENID
O[SPC]UN[SPC]PREMIODE:"W"PUNTOS"
666 POKES2,217:FORT=1TO300:NEXT:POKE
S2,0:POKES2,213:FORT=1TO300:NEXT:POK
ES2,0:POKES2,223
667 FORT=1TO300:NEXT:POKES2,0:POKES2
,227:FORT=1TO150:NEXT:POKES2,0:POKES
2,234
689 FORL=1TO100*M:NEXT
690 POKE36874,0:POKE36875,0:POKE3687
6,0:FORL=1TO40:NEXT:NEXT
691 POKE36878,15:FORL=202TO238STEP.8
:POKE36876,L:NEXT:POKE36876,0:FORL=1
TO10:NEXT
692 FORL=155TO228STEP2:POKE36876,L:N
EXT:FORL=228TO135STEP-2:POKE36876,L:
NEXT:POKE36876,0
695 REM AVANCE
1005 GOTO1160
```



```
1010 PRINT"[HOM][RVSON][BLK][18CRSRD]
[7CRSRR]AVANCE[CRSRD][8CRSRL]PULSE[SPC]
SPACE"
1011 GETQ$:IFQ$=""THEN1010
1015 IFQ$="N"THEN1047
1020 IFQ$="[SPC]"THENGOSUB1110:GOTO1
040
1021 GOTO1010
1040 PRINTTAB((N-1)*7+1)"[SPC]"
1046 GOTO1049
1047 PRINTTAB((N-1)*7+1)"[SPC]"
1049 NEXTN
1110 A=INT(RND(1)*8)+1
1120 A(N)=A
1130 PRINTTAB((N-1)*7+1)""
1140 PRINT"[HOM][4CRSRD][CRSRR]";TAB
((N-1)*7+1)A$(A)
1145 PRINT"[HOM]"
1150 IFA(1)=A(2)ANDA(2)=A(3)THENW=15
:GOTO665
1155 REM MENSAJE
1160 FORT=1TO2000:NEXT
1161 PRINT"[CLR][RVSON][2CRSRR][8CRSRD]
[BLK]SIGUE[SPC]JUGANDO?"
1162 PRINT"[HOM][RVSON][10CRSRD][2CRSRR]
[PUR]SI=[BLK]PULSE[SPC]UNA[SPC]TECLA
"
1163 FORT=1TO2000:NEXT:PRINT"[HOM][RVSON]
[2CRSRR][14CRSRD][PUR]NO=[BLK]APAGA[SPC]
Y[SPC]VAMONOS"
1169 GETQ$:IFQ$=""THEN1169
1170 RUN
668 FORT=1TO150:NEXT:POKES2,0:POKES2
,230:FORT=1TO300:NEXT:POKES2,0:POKES
2,227
669 FORT=1TO150:NEXT:POKES2,0:POKES2
,234:FORT=1TO150:NEXT:POKES2,0:POKES
2,230
670 FORT=1TO300:NEXT:POKES2,0:POKES2
,223:FORT=1TO300:NEXT:POKES2,0:POKES
2,227
```

```
671 FORT=1TO300:NEXT:POKES2,0:POKES2
,217:FORT=1TO300:NEXT:POKES2,0:POKES
2,213
672 FORT=1TO400:NEXT:POKES2,0
673 POKES2,217:FORT=1TO300:NEXT:POKE
S2,0:POKES2,213:FORT=1TO300:NEXT:POK
ES2,0:POKES2,223
674 FORT=1TO300:NEXT:POKES2,0:POKES2
,227:FORT=1TO150:NEXT:POKES2,0:POKES
2,234
675 FORT=1TO150:NEXT:POKES2,0:POKES2
,230:FORT=1TO300:NEXT:POKES2,0:POKES
2,227
676 FORT=1TO150:NEXT:POKES2,0:POKES2
,234:FORT=1TO150:NEXT:POKES2,0:POKES
2,230
677 FORT=1TO300:NEXT:POKES2,0:POKES2
,223:FORT=1TO300:NEXT:POKES2,0:POKES
2,227
678 FORT=1TO300:NEXT:POKES2,0:POKES2
,217:FORT=1TO300:NEXT:POKES2,0:POKES
2,213
679 FORT=1TO400:NEXT:POKES2,0:POKE36
875,0:POKE36878,0:GOTO1005
680 POKE36878,15:S2=36874:PRINT"[HOM]
[RVSON][BLK][15CRSRD]HAS[SPC]OPTENID
O[SPC]UN[SPC]PREMIO[SPC]DE:"W"PUNTOS
"
681 POKES2,225:FORT=1TO200:NEXT:POKE
S2,0:POKES2,232:FORT=1TO200:NEXT:POK
ES2,0:POKES2,237
682 FORT=1TO200:NEXT:POKES2,0:POKES2
,240:FORT=1TO300:NEXT:POKES2,0:POKES
2,237
683 FORT=1TO200:NEXT:POKES2,0:POKES2
,240:FORT=1TO350:NEXT:POKES2,0
688 POKE36878,15:FORM=1TO8STEP7:POKE
36876,210:POKE36875,210:POKE36874,21
0
```



PROGRAMA RENUMERADOR

C-64

Queridos amigos: Os adjunto un pequeño programa que no dudo tendrá buena acogida entre los muchos aficionados.

Es una pequeña subrutina renumeradora que considero muy práctica a pesar de tener el inconveniente de no renumerar la nueva dirección que toman GOTO, GOSUB o THEN, de todas formas es muy útil en la mayoría de los casos y siempre podemos señalizarnos los saltos con una sentencia REM. El programita se puede cargar al principio o sobre la marcha, ya que es lo bastante corto para que nadie se aburra tecleando.

A pesar del inconveniente indicado creo es bastante útil pues a pesar de que todos utilizamos la buena táctica de dejar espacios de separación entre líneas que consideramos suficientes, al final el programa crece y crece y se nos desborda. Espero poder enviar pronto las sentencias que nos permitan renumerar los GOTO, etc., o si alguno lo ha conseguido, ánimo y envia tu programa a la revista, nos podemos ayudar y beneficiarnos entre todos intercambiando nuestras experiencias.

Ya sólo un pequeño comentario sobre

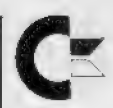


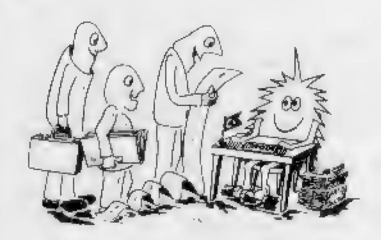
el programa, la Base en este caso corresponde a la dirección de comienzo del basic del C-64, deberéis hacer el cambio oportuno para el VIC-20. El tamaño de la Matric está relacionado con la cantidad de sentencias del programa y la separación entre ellas. En la línea 63970 imprimimos en pantalla la línea original y su correspondiente nueva, podemos eliminar la parte de variables a imprimir si queremos más rapidez aunque os aseguro que no es nada lento.

Hasta pronto,

Domingo Márquez Gallego
Parque Vosa, 3-4º C.
Móstoles (Madrid). Teléf.: 614 21 48

```
63992 REM*****************************
63994 REM*                           *
63995 REM* PROGRAMA RENUMERADOR      *
63996 REM* AUTOR DOMINGO MARQUEZ     *
63997 REM* ABRIL 1984 ** C-64        *
63998 REM*                           *
63999 REM*****************************
63900 CLR:BASE=2048:LA=BASE
63910 INPUT"[illegible]
[illegible]
63920 DIM[illegible](1000),[illegible]
63930 NU=PEEK(LA+1)*256+PEEK(LA+2)
63940 NN=PEEK(LA+3)*256+PEEK(LA+[illegible]
63950 IF[illegible]THEN 6399[illegible]
63960 POKE(LA+3),(I) AND 255:POKE(LA+
[illegible],I/256)
[illegible]
[illegible]
[illegible]
[illegible]
```



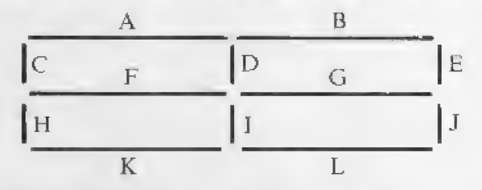


ASTUCIA

VIC-20

Es muy sencillo de explicar, consiste en que el VIC-20 está programado para hacer preguntas.

Comienza con una espectacular presentación y luego pasa a hacer la primera pregunta. La pregunta ya dice lo que debemos hacer. La primera es difícil de entender en lo que dice de la letra de cada palillo. Lo que dice es esto:



Luego pasa a hacer las siguientes que ya se entienden. Al final dice los puntos totales y acaba.

Si se precisa de ampliación de memoria, cosa que yo no, se pueden poner muchas más preguntas.

Antonio Font (15 años)
C/Mañe y Flaquer, 18, 2º 1º
Barcelona
Teléfono: 203 15 10
Estudiante de F.P. de Electrónica 1.er grado, 2º curso.

N. de la R.: Los otros programas que nos has enviado, serán publicados en números sucesivos.

```
10 PRINT"[CLR]"
11 GOSUB1060:GOSUB1060
15 PRINT"[5CRSRR][6CRSRD]"
20 PRINT"[RVSON]1-PREGUNTA[RVSOFF]"
24 FORZ=1TO1000:NEXT
25 PRINT"[CLR]"
30 PRINT:PRINT"[RVSON]FUGA[SPC]DE[SPC]
CUADROS[RVSOFF]"
40 PRINT:PRINT"RETIRANDO[SPC]DE[SPC]
LA[SPC]FIGU-[SPC]RA[SPC]SUPERIOR[SPC]
SOLO"
50 PRINT"DOS[SPC]PALILLOS,LOS[SPC]CU
-[2SPC]ATRO[SPC]CUADROS[SPC]SE[SPC]C
ON-[2SPC]VIERTEN[SPC]EN[SPC]DOS"
60 PRINT:PRINT"-PARA[SPC]ELIMINAR[SPC]
UN[SPC]PALI[SPC]LLO[SPC]PULSA[SPC]LA
[SPC]LETRA[4SPC]CORRESPONDIENTE[SPC]
HASTA[SPC]LA[SPC]L"
65 PRINT"[SPC](SIGUIENDO[SPC]EL[SPC]
ABECE-[2SPC]DARIO)"
70 PRINT"-PARA[SPC]VOLVER[SPC]A[SPC]
VER[SPC]LA[2SPC]FIGURA[SPC]INUCIAL[SPC]
PULSA[2SPC]0"
71 FORZ=1TO20000:NEXT
75 PRINT"[CLR][4CRSRD]"
81 PRINT:PRINT"-LOS[SPC]NUMEROS[SPC]
VAN[SPC]POR[3SPC]CADA[SPC]PALILLO[SPC]
HORIZON-[SPC]TALMENTE[SPC]Y[SPC]EN[SPC]
ORDEN"
82 PRINT"-CUANDO[SPC]ESTE[SPC]SEGURO
[SPC]DE[SPC]LA[SPC]SOLUCION[SPC]PULS
E[5SPC]SPACE"
83 FORZ=1TO8000:NEXT
90 GOSUB400
91 W=0
150 PRINT"[CLR]"
```

```
155 W=W+1
160 PRINTA$;B$:PRINTE$;J$;I$:PRINTC$
;D$:PRINTF$;K$;L$:PRINTG$;H$
168 IFZ$="[SPC]"THEN250
169 IFW=5THEN250
170 GETZ$:IFZ$=""THEN170
171 IFZ$="A"THENA$="[3SPC]":GOTO150
172 IFZ$="B"THENB$="[4SPC]":GOTO150
173 IFZ$="C"THENE$="[SPC][CRSRL][CRSRD]
[SPC][CRSRL][CRSRD][SPC]":GOTO150
174 IFZ$="D"THENJ$="[2CRSRU][3SPC][CRSRD]
[CRSRL][SPC][CRSRD][CRSRL][SPC]":GOT
O150
175 IFZ$="E"THENI$="[2CRSRU][3SPC][CRSRD]
[CRSRL][SPC][CRSRD][CRSRL][SPC]":GOT
O150
176 IFZ$="F"THENC$="[3SPC]":GOTO150
177 IFZ$="G"THEND$="[4SPC]":GOTO150
178 IFZ$="H"THENF$="[SPC][CRSRL][CRSRD]
[SPC][CRSRL][CRSRD][SPC]":GOTO150
179 IFZ$="I"THENK$="[2CRSRU][3SPC][CRSRD]
[CRSRL][SPC][CRSRD][CRSRL][SPC]":GOT
O150
180 IFZ$="J"THENL$="[2CRSRU][3SPC][CRSRD]
[CRSRL][SPC][CRSRD][CRSRL][SPC]" GOT
O150
190 IFZ$="K"THENG$="[3SPC]":GOTO150
200 IFZ$="L"THENH$="[5SPC]":GOTO150
210 IFZ$="0"THENGOTO90
220 GOTO150
250 PRINT"[CLR]"
260 PRINT"[6CRSRD]"
270 PRINT"[RVSON]1-SOLUCION[RVSOFF]"

280 FORZ=1TO1000:NEXT
290 PRINT"[CLR]"
```



```
300 GOSUB400
301 C$="[3SPC]":J$="[2CRSRU][2SPC][CRSRD]
[SPC][CRSRD]"
310 PRINTA$,B$:PRINTE$;J$;I$:PRINTC$
,D$:PRINTF$,K$;L$:PRINTG$,H$
320 FORZ=1TO5000:NEXT
330 GOTO550
400 A$="[3COMM@]":B$="[4COMM@]"
410 C$="[RVSON][3SPC][RVSOFF]":D$="[RVSON]
[4SPC][RVSOFF][COMMH]"
420 C$="[RVSON][3SPC][RVSOFF]":D$="[RVSON]
[4SPC][RVSOFF][COMMH]"
430 E$="[COMMG][CRSRL][CRSRD][COMMG]
[CRSRL][CRSRD][COMMG]"
440 F$=E$
450 G$="[3COMMT]":H$="[4COMMT]"
460 I$="[2CRSRU][3SPC][COMMG][CRSRD]
[CRSRL][COMMG][CRSRD][CRSRL][COMMG]"
470 L$=I$
480 J$="[2CRSRU][2SPC][SHIFTB][CRSRD]
[CRSRL][SHIFTB][CRSRD][CRSRL][SHIFTB]
"
490 K$=J$
500 RETURN
550 PRINT"[CLR][6CRSRD]"
560 PRINT"[RVSON]2-PREGUNTA[RVSOFF]"

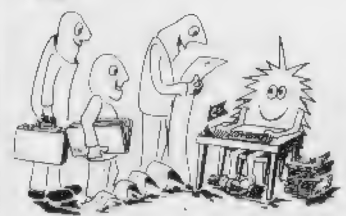
570 FORZ=1TO1000:NEXT
580 PRINT"[CLR]"
590 PRINT"[RVSON]EL[SPC]NUMERO[SPC]Q
UE[SPC]NO[SPC]DEJA[SPC][CRSRD]FRACCI
ONES[RVSOFF]"
600 PRINT:PRINT"CUAL[SPC]ES[SPC]EL[SPC]
NUMERO[SPC]QUE[SPC][CRSRD]SE[SPC]PUE
DE[SPC]DIVIDIR"
610 PRINT"[CRSRD]EXACTAMENTE[SPC]POR
[SPC]TODOS[SPC][CRSRD]LOS[SPC]DIGITO
S,DEL[SPC]1[SPC]AL[SPC][CRSRD][SPC]9
,AMBOS[SPC]INCLUSIVE?"
611 PRINT:PRINT"SI[SPC]QUIERE[SPC]VE
R[SPC]LA[SPC]SO-[2SPC]LUCION[SPC]PUL
SE[SPC]0"
620 PRINT:PRINT:INPUTN
621 X=50
622 IFN=0THENGOTO642
630 IFN=2520THENINPUT"COMO[SPC]LO[SPC]
HAS[SPC]HECHO";C$
640 IFC$="9*8*7*5"THENPRINT"ERES[SPC]
BUEN[SPC]MATEMATICO":X=X+20:GOTO700
641 GOTO580
642 PRINT"[CLR][6CRSRD][RVSON]2-SOLU
CION[RVSOFF]"
643 FORZ=1TO1000:NEXT:PRINT"[CLR]ES[SPC]
EL[SPC]2520"
644 PRINT"[CRSRD]PORQUE[SPC]ES[SPC]9
*8*7*5=2520"
645 FORZ=1TO8000:NEXT:GOTO720
650 PRINT"FALLASTE":FORZ=1TO1000:NEX
T:GOTO580
700 FORZ=1TO1000:NEXT
720 PRINT"[CLR][6CRSRD]"
730 PRINT"[RVSON]3-PREGUNTA[RVSOFF]"

740 FORZ=1TO1000:NEXT
750 PRINT"[CLR]"
760 PRINT"[RVSON]EL[SPC]VASO[SPC]DE[SPC]
AGUA[RVSOFF]"
770 PRINT:PRINT"COMO[SPC]COLOCAR[SPC]
BOCA[5SPC][CRSRD]ABAJO[SPC]UN[SPC]VA
SO[SPC]LLENO[SPC]DE[SPC][CRSRD]AGUA[SPC]
SIN[SPC]DERRAMAR[SPC]NI"
780 PRINT"[CRSRD]UNA[SPC]GOTA"
781 PRINT"SI[SPC]QUIERES[SPC]VER[SPC]
LA[SPC]SO-[SPC]CION[SPC]PULSA[SPC]0"
790 PRINT"[CRSRD][RVSON]CON[SPC]UNA[SPC]
PALABRA[SPC]BASTA[RVSOFF]":INPUT"SE[SPC]
";P$
791 IFP$<>"0"THENGOTO800
792 PRINT"[CLR][6CRSRD][RVSON]3-SOLU
CION[RVSOFF]":FORZ=1TO1000:NEXT:GOTO
820
800 IFP$<>"CONGELA"THENGOTO750
810 PRINT:PRINT"MUY[SPC]BIEN":X=X+20
:FORZ=1TO8000:NEXT:GOTO850
820 PRINT"[CLR][SPC][3CRSRD]SE[SPC]C
ONGELA"
830 FORZ=1TO8000:NEXT
850 PRINT"[CLR][6CRSRD][RVSON]4-PREG
UNTA[RVSOFF]"
860 FORZ=1TO1000:NEXT
870 PRINT"[CLR][2CRSRD][RVSON]EL[SPC]
GUARDARROPAS[RVSOFF]"
880 PRINT:PRINT"TODAS[SPC]LAS[SPC]CA
MISAS[SPC]SON[SPC][CRSRD]BLANCAS,MEN
OS[SPC]2;"
890 PRINT"[CRSRD]TODAS[SPC]SON[SPC]A
ZULES,[5SPC][CRSRD]MENOS[SPC]2;"
900 PRINT"[CRSRD]TODAS[SPC]SON[SPC]R
OSADAS,[4SPC][CRSRD]MENOS[SPC]2."
910 PRINT"[CRSRD]CUANTAS[SPC]CAMISAS
[SPC]HAY[3SPC][CRSRD]DE[SPC]CADA[SPC]
COLOR?"
920 PRINT"[CRSRD]PARA[SPC]VER[SPC]LA
[SPC]SOLUCION[SPC]PULSA[SPC]0"
930 INPUTW:IFW=0THENGOTO960
940 IFW<>3THENPRINT"MUY[SPC]MAL":FOR
Z=1TO800:NEXT:GOTO870
950 PRINT"MUY[SPC]BIEN":FORZ=1TO8000
:NEXT:X=X+20:GOTO990
960 PRINT"[CLR][6CRSRD][RVSON]4-SOLU
CION[RVSOFF]"
970 FORZ=1TO1000:NEXT
980 PRINT"[CLR][2CRSRD]SON[SPC]3:POR
QUE[SPC]HAY[SPC]UNA[3SPC][CRSRD]AZUL
,UNA[SPC]BLANCA[SPC]Y[4SPC][CRSRD]UN
A[SPC]ROSADA"
990 PRINT"[CLR][5CRSRD]HAS[SPC]CONSI
GHIDO";X;"PUNTOS"
991 FORZ=1TO5000:NEXT
1000 PRINT"[CLR][8CRSRD]"
1010 PRINT"[SHIFTB][COMMT][SPC][SHIFTO]
[SPC][SHIFTG][SHIFTM][3SPC][SHIFTY]"

1020 PRINT"[COMMQ][SHIFTC][SPC][COMM+]
[SPC][SHIFTG][SPC][SHIFTM][2SPC][SHIFTY]
"
1030 PRINT"[SHIFTB][2SPC][COMM+][SPC]
[SHIFTG][2SPC][SHIFTM][SPC][SHIFTH]"

1040 PRINT"[SHIFTB][2SPC][COMM+][SPC]
[SHIFTG][3SPC][SHIFTM][SHIFTH]"
1041 FORZ=1TO8000:NEXT
1050 PRINT"[CLR]":GOSUB1060:GOSUB106
0:PRINT"[CLR]":END
1060 POKE36878,15
1070 FORL=1TO100
1075 C=INT(RND(1)*255)+1
1080 POKE36879,C
1090 POKE36876,INT(RND(1)*128)+128
1100 FORM=1TO10
1110 NEXTM:NEXTL
1120 POKE36876,0:POKE36878,0:POKE368
79,27
1130 RETURN
```





MEMORIA FIJA PARA EL VIC-20

Alfredo Sindin Valero
Avda. Logroño, 42. Villa Esther
Madrid-22

¡Hola a todos!

Me llamo Alfredo, tengo 18 años y soy uno de los más antiguos VIC-iosos de este país, pues hace ya año y medio que tengo un VIC-20.

Os envío en plan de colaboración tres programas: una utilidad y dos juegos, en cinta, pues a falta de impresora... En fin, paso a describirlos:

1.—VIC-Memoria; Este programa **debe** ser usado con la ampliación de 16K (ó 8 K). Consiste en una corta rutina en código máquina que, por un lado, mueve la memoria de pantalla a la posición 7680 (igual que en un VIC sin ampliación), y por tanto, coloca el inicio del BASIC por encima del fin de pantalla; quedan así libres 3,5 K de memoria (posiciones 4608 a 7679), que estando libres de cualquier intrusión del Basic, pueden ser utilizados para rutinas de código máquina, caracteres definidos por el usuario o pantallas alternativas.

Aconsejo desensamblar el código máquina y analizarlo, pues puede ayudarnos a comprender el funcionamiento del VIC-20.

```
0 REM  ******************         *VIC
MEMORIA*                    *[illegible]*
            ****************
10 DATA1[illegible],160 [illegible] 24 [illegible] 152 [illegible] 169
 [illegible] 141,128,2,169 240,141,5,144,17[illegible],
2,144,9,1[illegible],141
11 DATA2,144 169,30,141,136,2 1[illegible]
[illegible],141,132,2,169,0,141,15,144,169,1,1
41 134,2,[illegible] 1[illegible]
12 DATA[illegible],96
20 FORI=1[illegible]TO1[illegible]:READA:POKEI,A:N
EXT:SYS1[illegible]
```

Nota de redacción:

Las otras colaboraciones que nos has enviado las iremos publicando en los siguientes números de la revista, si las pusiésemos todas juntas te dejaríamos la revista para tí solo.

Gracias por tus colaboraciones y sigue adelante con tus programas...

GRAFICOS MATEMATICOS

Estimados amigos:

En el último ejemplar de COMMODORE WORLD, dejabais en manos de vuestros lectores el mandar algún programa que trate de la alta resolución. También en el 1º ejemplar explicabais un poco esto; pues habiendo hecho pruebas, me he encontrdo con unos resultados muy satisfactorios.

El programa se realiza en un área de 64×64 pixels ya que para controlar los 32384 pixels de toda la pantalla, son necesarios más de los 3,5 kb que tiene el VIC-20 standard.

Si no les importa y publicasen mi programa, les agradecería que publicasen este saludo a Paco Cancio, también un VIC-cioso.

El programa dibujará la gráfica de un seno.

Para dibujar el gráfico de la llamada campana de Gauss, hay que sustituir la línea 60 por la siguiente:

60 Y=INT (32+31*COS(X/10))

Para dibujar una elipse, habrá que añadir estas líneas:

55Y1=32+SQR (64*X-X*X)
56Y2=32-SQR (64*X-X*X)
60 FOR Y=Y1TOY2STEPY2-Y1
125NEXT

A continuación paso a describir el programa:

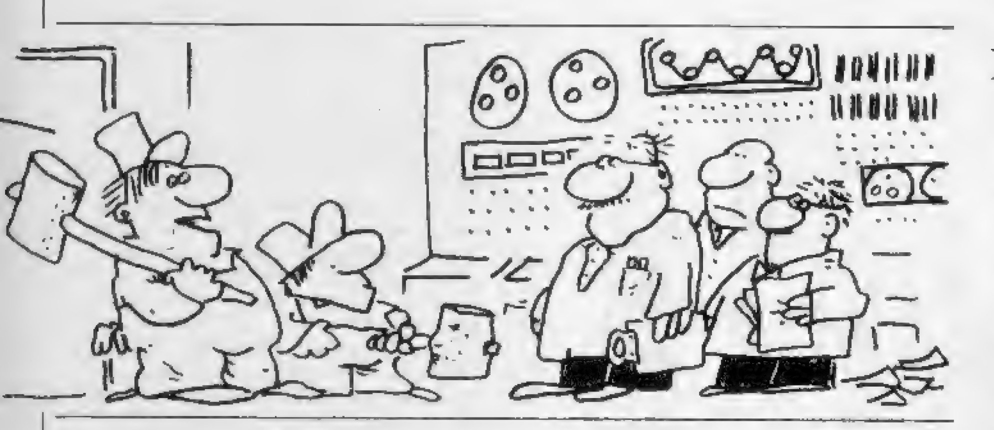
LINEA	COMENTARIO
5	PASA A CARACTERES PROGRAMABLES
6	PONE A CERO TODOS LOS PIXELS
10	PANTALLA NEGRA Y BORRAR PANTALLA
20	
30	VISUALIZAN LOS 64 CARACTERES DE LA FORMA 8 8
40	
50	FOR PARA LA COORDENADA X
60	NOS DA LA COORDENADA Y
70	NOS DA EL NUMERO DEL CARACTER EN EL CUAL SE VA A ENCENDER EL PIXEL
80	NOS DA EL CODIGO DE VISUALIZACION DE DICHO CARACTER
90	DOS DA EL BYTE DE DICHO CARACTER EN EL CUAL SE VA A ENCENDER EL PIXEL
100	NOS DA EL BIT DE DICHO BYTE QUE VA A SER ENCENDIDO
110	ENCIENDE ESE BIT, DENTRO DE ESE BYTE, QUE PERTENECE A DICHO CARACTER
120	NEXT PARA EL FOR DE X DESPUES, PULSEN STOP/RESTORE YA QUE LA LINEA 130 ES UN BUCLE CERRADO

```
1 REM****GRAFICOS*****
2 REM*JAVIER ELUSTONDO
3 REM*****PARA*******
4 REM*COMMODORE WORLD*
5 POKE36869,255
6 FORI=7168TO7679:POKEI,0:
NEXT
10 POKE36879,8:PRINTCHR$
(147)
20 FORL=0TO7:FORM=0TO7
30 POKE7680+M*22+L,L*8+M
40 NEXTM:NEXTL
50 FORX=0TO63
60 Y=INT(32+31*SIN(X/10))
70 CH=INT(X/8)*8+INT(Y/8)
80 RO=(Y/8-INT(Y/8))*8
90 BY=7168+8*CH+RO
100 BI=7-(X-(INT(X/8)*8))
110 POKEBY,PEEK(BY)OR(2↑BI)
120 NEXT
130 GOTO130
```

Javier Elustondo González-Pinto
C/ María Goiri, 3-2º Dcha.
Algorta (Vizcaya)
(14 años)
Tel.: 469 64 94

El Misterio de los Pokes

Por Diego ROMERO



Continuando en orden ascendente de dirección, este mes nos encontramos con las posiciones de memoria 3, 4, 5 y 6, que son sencillamente unos vectores que utiliza el intérprete BASIC para apuntar a las direcciones donde se encuentran en la ROM las rutinas de conversión de números en notación de coma flotante al de enteros y viceversa.

La forma en que se guardan estos vectores puede resultar un poco peculiar para aquellos que no estén familiarizados con el modo en que la CPU guarda las direcciones de los saltos en el modo de direccionamiento Indirecto Absoluto, pero la considerarán lógica todos los que conozcan los modos de direccionamiento de nuestra CPU. Vamos a explicarla un poco:

El sistema basado en las CPUs de la serie 6500, utiliza un bus de direcciones de 8 líneas, por lo que puede almacenar en la memoria números comprendidos entre 0 y 255; para almacenar cifras mayores se ve obligado a partirlas en dos bytes, uno de mayor peso o byte alto y otro de menor peso o byte bajo, con ellos puede manejar cifras entre 0 y 65535, con lo que ya puede apuntar a cualquiera de sus posiciones de memoria, pero para guardarla como un vector debe colocarla en dos posiciones consecutivas de memoria en un orden fijado al diseñar el sistema, que es el siguiente:

—El byte de menor peso se guarda en la primera dirección de memoria.

—El byte de mayor peso se guarda en la segunda dirección de memoria.

De este modo están guardados los vectores de las posiciones que estamos estudiando, el vector almacenado en las direcciones 3 y 4 apunta a la rutina que convierte los números de coma flotante a enteros, la parte menos significativa o de menor peso se guarda en la posición 3 y la más significativa en la 4.

Las direcciones 5 y 6 apuntan a la rutina que convierte los enteros a la notación de coma flotante.

Si deseamos alterar las rutinas normales de conversión para utilizar una notación peculiar nuestra, debemos modificar el contenido de estas posiciones de memoria para que apunten a nuestra rutina en código máquina.

Las posiciones de memoria siguientes son una serie de punteros y registros de almacenamiento temporales que utiliza el intérprete basic para buscar caracteres, guardar la columna a la que apunta TAB, etc., y nosotros no debemos alterarlos para evitar interferir el correcto funcionamiento del BASIC. Incluimos una lista con los nombres y funciones de cada una de estas posiciones de memoria en la tabla:

En el VIC y el C-64 los vectores son normalmente:

	3	4	5	6	C. Flot. a enteros	Enteros a C. Flot
VIC-20	$AA	$D1	$91	$D3	$D1AA	$D391
C-64	$AA	$B1	$91	$B3	$B1AA	$B391

Dirección Hex	Dec	Nombre	Función
7	7	CHARAC	Búsqueda de carácter.
8	8	ENDCHR	Bandera de búsqueda de comillas
9	9	TRMPOS	Columna de TAB.
A	10	VERCK	Flag. O=LOAD. 1=VERIFY
B	11	COUNT	Puntero de entrada (Nº de caracteres).
C	12	DIMFLG	Dimensión por defecto de matriz
D	13	VALTYP	Tipo de datos: FF=String. 00=Numérico
E	14	INTFLG	Tipo de datos: 80=Enteros. 00=C. Flotante
F	15	GARBFL	Bandera múltiple (comillas/list/etc.).
10	16	SUBFLG	Bandera subíndice/función (FN)
11	17	INPFLG	B. de entrada 00=INPUT 40=GET/98=READ
12	18	ATNSGN	ATN/signo/compara resultado
13	19	PROMTF	Bandera de I/0 actual
14	20	INTNUM	Número entero
15	21	INTNUM	Número entero
16	22	TEMPPT	Puntero temporal de strings
17	23	LASTPT	Puntero último literal
19	25	TEMPST	Stack Temporal de literales
22	34	INDEX	Indices de utilidad.
26	38	RESHO	Zona trabajo producto de multiplicación

En el próximo número veremos la función de los punteros de principio y fin de memoria que alteramos en muchos programas Basic para utilizar caracteres personalizados y protegerlos de la posible destrucción con variables del programa...

Magia



La MAGIA son trucos, la MAGIA es divertida.

La MAGIA es hacer lo que nadie se ha atrevido.

La MAGIA es una columna mensual llena de consejos, trucos, de esto y aquello del mundo del software, hardware y aplicaciones.

Cada mes, MAGIA les trae trucos breves y útiles de informática procedentes de todo el mundo - trucos descubiertos por los demás que hacen que la informática sea más fácil, más divertida o más animada.

MAGIA habla de ideas sencillas, programas de una sola línea, subrutinas útiles, hechos de informática poco conocidos y otras cosas de interés. Buscamos material nuevo o renovado que resulta ser de valor actual para usuarios de equipos Commodore y que puede utilizarse con un mínimo de tiempo, esfuerzo o conocimientos teóricos.

MAGIA resulta ser la fuente más completa de información para la informática práctica, además de ser un foro internacional para compartir trucos con otros aficionados. Envíe sus trucos a:

COMMODORE WORLD
Pedro Muguruza, 4
Madrid-16

La revista "Commodore World" sorteará seis paquetes de software en julio y diciembre entre todas las contribuciones publicadas.

LISTADO EN COLORES

Conecte el VIC y prepare la siguiente línea:

10 REM "M[CTRL RED]"

Pulse RETURN. Llévese el cursor sobre la letra M y Pulse [CTRL RYSON] y a continuación [SHIFT M].

Pulse RETURN de nuevo, y donde había M deberá aparecer ahora una barra diagonal en reversa.

Teclee unas líneas cualesquiera (20 PRINT... 30 REM... etc). Pida LIST y se listará, si todo ha ido bien, en el color que hubiera colocado en el REM de la línea 10.

Colocar una linea así al principio de cualquier programa, provocará el listado en el color que se desee. Si el color elegido es el mismo del fondo, logrará el listado invisible.

Además todos los caracteres de control en vídeo inverso que se pueden obtener desde el teclado, [HOME], [CLR], [CRSRS], etc, funcionarán con este Procedimiento al hacer un listado por pantalla.

Francesc Castanyer P.
C/Queretano, 5, E - 2ª, 2º D
Palma de Mallorca, 7
Baleares

• • •

VIC-20

Este es un truco muy sencillo, pero que en ocasiones no deja de tener su importancia, ahí va.

Si quieres evitar el acceso de los demás a tus programas y que sólo puedan ser utilizadas por personas allegadas, te aconsejo colocar como primera línea:

10 POKE 775, 200: INPUT "DAME CLAVE"; C$: IFC$ <> D$ THEN 63999

(siendo D$ cualquier clave que introduzcas).

Y pon como última línea;

63999 PRINT "LO SIENTO, OTRA VEZ SERA": SYS 64802.

En la línea 10, el POKE 755, 200 es para impedir el listado del programa y así poner a salvo tu clave. La línea muestra el rótulo entre comillas y después simula la desconexión de tu VIC (ante el asombro del intruso).

Si quieres, en vez de una clave, entra un nombre pero si es compuesto o una abreviación del mismo como por ejemplo JAVI/JAVIER O JOSE MARI/JOSE MARIA debes sustituir en la línea 10; 10 POKE 755, 200: INPUT "DAME TU NOMBRE"; C$: LEFT$ (C$, 4) <> D$ THEN 63999

(siendo D$=JAVI).

Por que por una letra más podrias tener que volver a cargar de nuevo el programa y eso no es agradable.

¡Ah! y por último, una vez que esté ejecutando el programa vigila que no llegue a la linea 63999 (puedes colocar un END en la línea anterior).

No es egoísmo, simplemente un derecho. GRACIAS.

José María Rodriguez
Plaza del Puente, 7-4º dcha.
Las Arenas
VIZCAYA.

• • •

"QUITAR PROTECCIONES"

Al comprar cintas ya programadas, puede que el comprador quiera estudiar dicho programa además de ejecutarlo; pero al mandar al ordenador listarlo, nos encontramos con algún error de este tipo: (Las cifras e instrucciones pueden variar).

Ejemplo 1:
LIST
10SYS9876: REM
SYNTAX
ERROR
READY

Ejemplo 2
LIST
10 REM
SYNTAX
ERROR
READY

Esto lo solucionamos de este modo: Tecleamos

Ejemplo 1:
10SYS9876
LIST

Ejemplo 2:
10
LIST

Una vez realizado esto nos aparecerá el listado.

Javier Elustondo Glez-Pinto.
C/María Goiri, 3-2º dcha.
Algorta (Vizcaya). Teléf.: 469 64 94
Edad: 13 años.

• • •

CONTROL DE PANTALLA

Amigos de Commodore:

Os envío estas líneas esperando que aparezcan en vuestra sección de MAGIA. Se trata de unos POKES que pueden resultar interesantes (Para VIC-20).

El primero de ellos es el 646, cuya función es la de cambiar el color del cursor. Las posiciones de memoria correspondientes a cada color son una unidad menor que el número que aparece encima de cada color del cursor en el teclado, es decir: 0 para el negro; 1 para el blanco; 2 para el rojo; etc...

El segundo POKE está en la dirección de memoria 656. Este POKE, en la posición de memoria 128, vuelve la pantalla a su estado inicial.

Otro es el 36885, siendo una de sus funciones la de pasar a mayúsculas (240) o a minúsculas (posición 242).

Los siguientes POKES están relacionados con la posición de la pantalla en el marco. El 36864 controla la posición horizontal, y el 36865, la vertical, siendo sus estados normales las posiciones de memoria 12 y 38 respectivamente.

Los POKES 36866 y 36867 controlan el tamaño horizontal y vertical de la pantalla. Sus posiciones normales son las 12 y 46 respectivamente.

Leandro Olalla Martínez
Eugenio Selles 3, 5ºB (Pedregalejo)
Málaga-17

• • •

FACILITANDO LA EDICION

La magia trata sobre la utilización de la tecla (CTRL), que pulsada a la vez con otra tecla es equivalente a, por ejemplo, mover el cursor hacia abajo. Cuando pulsamos (CRSRD) entre comillas, vemos que nos escribe una "Q"; pues bien, (CTRL)+"Q" es equivalente a (CRSRD). Los resultados que se pueden obtener con esta tecla son:

(CTRL)+"£" .... (CTRL)+"3"
(CTRL)+"Q" .... (CRSRD)
(CTRL)+"E" .... (CTRL)+"2"
(CTRL)+"R" .... (CTRL)+"9"
(CTRL)+"T" .... (DEL)
(CTRL)+" " .... (CTRL)+"6"
(CTRL)+"S" .... (HOME)
(CTRL)+";" .... (CRSRR)
(CTRL)+"−" .... (CTRL)+"7"
(CTRL)+"N" .... (COMM)+(SHIFT) Sólo pasa a minúsculas.
(CTRL)+"M" .... (RETURN)

(CTRL)+"N", cuando queramos pasar a mayúsculas, hay veces que no se puede hacer con (COMM)+(SHIFT); lo podemos hacer con el CHR$ correspondiente. (CTRL) además nos escribe todas las letras en carácter inverso cuando escribimos entre comillas.

Y esto es todo. Si os parece que está mal redactado en alguna parte o lo queréis resumir, no tenéis más que cambiar lo que creáis conveniente.

José Luis Arostegui Plaza
Almona del Boquerón, 15
Granada-1

• • •

APROVECHANDO EL SUPEREXPANDER

En algunas ocasiones los usuarios del cartucho "Super Expander + 3K" no deseamos más que aprovechar los 3K de memoria RAM que contiene, pero nos encontramos con que el cartucho intercepta la ROM del VIC y "roba" algo de esos 3K. Cuando se conecta el VIC, con el cartucho conectado a él, el mensaje de la pantalla indica: 6519 BYTES FREE, lo que puede comprobarse por el contenido de las localizaciones 55 y 56 que son los valores 120 y 29, respectivamente. Es decir, faltan 136 bytes, que son los "secuestrados".

Pero existe una forma de hacer desaparecer toda la influencia del cartucho, quedándose únicamente con los 3K de RAM adicional.

Para ello escribir sencillamente:

POKE 256,120: POKE 257,76: POKE 258,47: POKE 259,253: SYS 256.

en forma directa, o con un número de línea.

Cuando esta orden se ejecuta aparecerá el mensaje:

6655 BYTES FREE

¡Ya están todos! Puede comprobarse que las localizaciones 55 y 56 contienen ahora los valores 0 y 30 respectivamente.

Hay que tener en cuenta que esta orden borra todo lo que existiese previamente en la memoria, así que es mejor hacerlo antes de comenzar a programar.

¿Magia?, bien nada es mágico si se conoce por qué funciona.

Atentamente.

Luis Suárez Antuña
C/Natividad, 2, 4º
Alcalá de Henares.

¡Cuidado! con este truco las teclas de función... ya no funcionan...!!!

• • •

PAUSAS EN PROGRAMAS

Hola amigos de Commodore World:

Os envío un truco para la sección de Magia sobre el VIC-20. Si queremos que un programa se pare hasta que una determinada tecla sea pulsada nosotros solemos escribir:

10 GET A$: IF A$<>"X" THEN 10

Esto lo podemos hacer ocupando menos memoria escribiendo:

10 WAIT 197, N

Donde N es específico de cada tecla "X". Para saber que N corresponde a cada tecla, podemos utilizar este corto programa:

20 PRINT "[Home]" PEEK (197): GOTO 20

Un RUN-20, lo hace correr y al mantener pulsada la tecla deseada, aparecerá en la esquina superior izquierda, el número correspondiente a esa tecla (mientras se mantenga pulsada).

Ej.: 10 WAIT 197,15

Espera a que se pulse la tecla RETURN.

Aquí os doy algunos valores:

SPACE 32	A 17	C 34
RETURN 15	B 35	= .... 46, etc.

Me despido ya de vosotros no sin antes felicitaros por el nivel y calidad de la revista, superior a otras que acaban de aparecer sobre los Commodore.

Miguel Angel Folgado Costa
Avda. Blasco Ibáñez, 29-1º
Apartado Correos, 19. Manises (Valencia)

• • •

AUTORREPETICION

Soy un lector de vuestra revista, y poseedor de un CBM 2001 y un Commodore 64. Me solidarizo con lo que dijo Armenio González Pérez en la sección CARTA BLANCA en el número 3 de abril, en relación a los usuarios de los primeros aparatos Commodore. Os envío esta pequeña rutina, que corre sólo en los ordenadores 2000 y 3000 con el BASIC 2, y que proporciona autorrepetición en todas las teclas. El programa en lenguaje máquina se carga en el buffer de la 2ª cassette, de manera que deja trabajar a la primera. Pero ATENCION!, si utilizamos la cassette número 2 nos arriesgamos a mandar todo a la porra.

```
10 DATA 169, 255, 197, 151, 208, 5, 169, 10, 141, 58, 3
20 DATA 206, 58, 3, 208, 11, 133, 151, 169, 5, 141, 58
30 DATA 3, 169, 3, 133, 168, 76, 46, 230
40 FOR I=827 TO 856: READ J: POKE I, J:NEXT I
50 POKE 144, 59: POKE 145, 3: NEW
```

Marco Rossi Heras
Calle Masferrer, 14, 1º
Barcelona-28

NOTA: Después de correr el programa es posible que el cursor desaparezca o se quede "tieso", pero se soluciona pulsando cualquier tecla.



DE TOMO Y LOMO

Como prometimos en el número anterior hemos estudiado más detenidamente los libros franceses sobre los PET/CBM.

El más elemental de ellos es "La decouverte du PET/CBM", es una toma de contacto con los ordenadores, sus instrucciones y comandos elementales, las bases de la programación, desarrollo de programas, obtención de gráficos y dibujos animados, y comunicación del PET con el mundo exterior. En los apéndices del libro encontramos tablas de las funciones y operadores BASIC, seguidos de un conjunto de ejercicios de programación y sus soluciones respectivas.

La practique du PET/CBM, está dividido en dos volúmenes, el primero dedicado a describir los periféricos y la gestión de ficheros, y el segundo al lenguaje máquina y ensamblador del 6502.

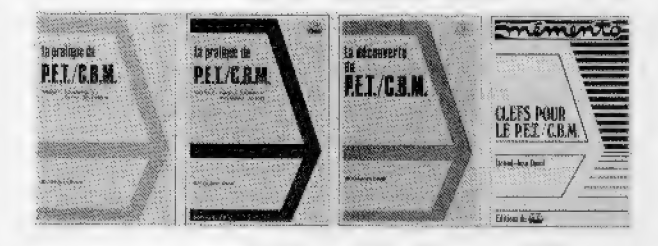
El primero detalla el almacenamiento de programas y datos en el casete de commodore, la utilización de la impresora para usos normales (texto) y gráficos, la unidad de disco CBM 3040, su manejo, gestión de bases de datos y el bus IEEE 448.

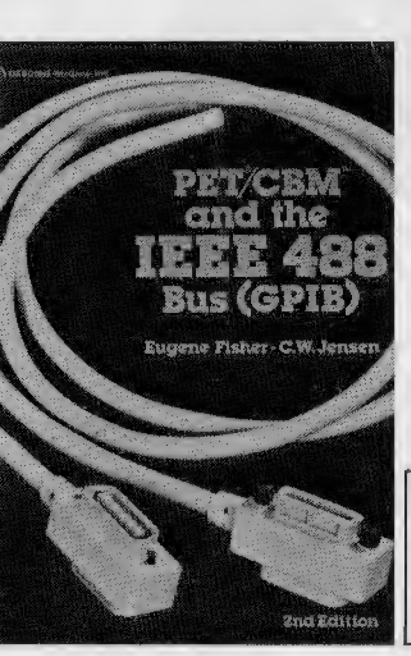
El segundo volumen es un análisis y descripción exhaustiva de la programación en lenguaje máquina, el editor, ensamblador y rutinas útiles del PET. Uno de los capítulos muestra el modo de intercambiar parámetros entre BASIC y código máquina por medio de las funciones SYS y USR.

El último de ellos, "Clefs pour le PET/CBM", es una guía rápida de tablas de utilidad, comandos, caracteres, mensajes de error, lenguaje máquina, patillajes de los conectores y circuitos integrados utilizados, mapa de memoria y trucos similares a los de la sección "Magia" de nuestra revista.

Todos los libros llevan ejemplos sencillos de rutinas en cada uno de los temas tratados, y pese a estar escritos en francés se puede entender su significado sin saber este idioma.

Los precios aparecieron en el número anterior de nuestra revista, y los distribuye P.S.I. Ibérica, Ferraz, 11-3, Madrid-8, teléfono: (91) 247 30 00.





Este libro es una guía completa sobre el bus de comunicaciones IEEE 448 utilizado en los ordenadores personales PET/CBM.

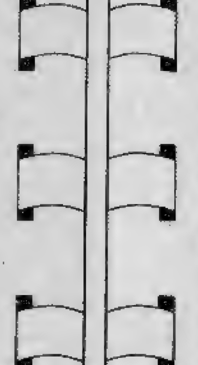
En los primeros capítulos da una serie de conceptos sobre comunicaciones entre ordenadores y periféricos, estructura del bus, líneas de control y de datos, protocolos de intercambio de información con diagramas de flujo detallados, líneas utilizadas por el IEEE 448 implementado en los equipos Commodore.

Más adelante aparecen diagramas de tiempos detallados de las señales al ejecutar las distintas secuencias de transferencia de datos desde BASIC, esquemas eléctricos de interfaces para utilizar periféricos IEEE de otras casas con los equipos Commodore, impresoras Centronics, etc.

En los últimos capítulos presenta varios ejemplos de control de aparatos por medio del bus (un analizador lógico, un circuito cerrado de televisión, etc.), una lista de los dispositivos de distintos fabricantes que pueden ser conectados al bus, tablas y unos programas de utilidad para los PET/CBM.

PET/CBM and the IEEE 448 bus (GPIB)
Autores: Eugene Ficher y C. W. Jesen
Editorial Mc. Graw Hill - España. Sta. Beatriz, 4. Madrid-18
Teléfono: (91) 231 32 03
P. V. P.: 3.200 pesetas.

GLOSARIO



Coma flotante:
Notación utilizada para trabajar con números reales en un ordenador (cj. 27,883779).

Flag:
Bandera o indicador. Bit de un registro que indica una condición determinada. En el caso de las CPUs 6502 y 6510, son los bits del registro de estado que indican cuándo el resultado de la operación anterior fue negativo, mayor o menor que otro, decimal o hexadecimal, cero, etc.

Indices:
Registros de la CPU utilizados para los modos de direccionamiento indexado.

Intérprete basic:
Programa encargado de "interpretar" los comandos y funciones de un programa BASIC y ejecutar las operaciones correspondientes a este comando o función.

LSB:
Least Significant Byte— Byte menos significativo de una dirección o palabra de varios bytes.

Mnemónicos:
Códigos utilizados para facilitar la programación en código lenguaje máquina; son abreviaciones de palabras normales en inglés, lo que facilita su memorización y utilización.

Modo de direccionamiento:
Técnica mediante la cual una instrucción se refiere a su operando. Los 6502 y 6510 tienen 13 modos de direccionamiento.

MSB:
More Significant Byte— Byte más significativo.

Notación entera:
Es la utilización para trabajar con números enteros comprendidos entre 32768 y -32767. Estos límites vienen dados por el modo en que el ordenador almacena en su memoria los números enteros (ocupando dos bytes).



Por motivos de espacio nos es imposible incluir en este número el listado de soft de Microelectrónica y Control. Continuaremos su publicación en el próximo número.

NOVEDAD

Microelectrónica y Control nos ha comunicado el lanzamiento del cartucho "SIMON'S BASIC", que añade 114 comandos Basic al C-64.

Entre los más importantes se encuentran: KEY, TRACE, INSERT, PLACE, CENTRE, PRINT, AT, FETCH, INKEY, MOD, DIV, PROC, IF...THEN...ELSE, COPY, CIRCLE, PAINT, DESIGN, MOB SET, DETECK, CHECK, WAVE, ENVELOPE, FLASH, etcétera.

Estamos esperando que Microelectrónica y Control nos envíe un cartucho de muestra para hacerles una amplia reseña, ya que se trata de un producto muy importante.

NOVEDADES PARA EL 64 EN EL CAMPO MEDICO

Nueva Batería de Aptitudes Psicomotoras (BAP-101)

Desarrollada por Metisistemas S.A. (Palma de Mallorca)

Desde principios de este siglo los doctores L.M. Patrizi, Loewenthal, W. Moede y C. Piorkowski entre otros, han estudiado la atención sostenida en los conductores de automóviles.

En España los primeros estudios fueron llevados a cabo en Barcelona en 1922 por el doctor E. Mira, quien realizó pruebas sobre velocidades y distancias, distribución, intensidad y duración de la atención visual, rapidez y precisión de la reacción visual y grado y tipo de la emotividad del sujeto. De estos estudios con los conductores de la compañía de autobuses, dedujo que se podían conseguir resultados notables con un examen corto, sin "jugar con la vida", siendo fácil escoger los mejores conductores y reducir con ello el número de accidentes.



Posteriormente han sido muchos los doctores que se han ocupado de la seguridad vial en distintos países, y en nuestro país, existe en la actualidad un equipo de tres personas que trabaja en este campo modernizando los medios de trabajo:

—D. José M.ª Sevilla Marcos. Médico Director del Centro de Reconocimientos Médico-Psicológicos para carnets de conducir.

—D. Antonio de la Hera Roldán. Psicólogo de Centro de Reconocimientos Médico-Psicológicos para carnets de conducir; y

—D. Emilio Medina Taltavull. Ingeniero, especialista en electrónica.

Estas tres personas han preparado diversos aparatos para la evaluación de la coordinación bimanual, y reacciones múltiples discriminatorias frente a estímulos visuales o sonidos.

Todos los instrumentos que han preparado son de gran utilidad para los exámenes de conducir, y dado el nivel tecnológico actual y la aparición de los ordenadores personales en el mercado nacional, han desarrollado un sistema computerizado para examinar conductores basado en la utilización de Commodore 64, la unidad de disco o cassette y la impresora.

Este equipo de profesionales ha diseñado unos periféricos especiales para esta aplicación, son: Controles manuales, pedales, periféricos para el control de proyectores de diapositivas, etc.

Para este tipo de aplicaciones han preparado programas para la evaluación de aptitudes psicomotrices en conductores y para la prevención de accidentes laborales.

Utilizando ordenadores personales, han conseguido máxima fiabilidad, menor número de errores en el manejo de resultados, procesarlos más rapidamente al hacerlo el propio programa y proporcionar los resultados por la impresora, menor número de averías que en los sistemas mecánicas tradicionales, menor coste y mayor rapidez en la realización de los exámenes al ahorrar los tiempos muertos inter-pruebas.

Para solicitar mayor información dirigirse a:

Metisistemas, S.A. Plaza Hornabeque, 4, 3º A. Palma de Mallorca. Tel.: (971) 23 45 63

CARTA BLANCA...

Deseo recibir información completa sobre la representación de funciones, aplicaciones matemáticas, etc. en el ordenador Commodore 64, ya que lo tengo y en las instrucciones que me dieron no explica muy bien cómo hacerlo.

DAVID ZARAGOZA DALMAN
ESTACION DE TAMARETE
LA MERLUZA (HUESCA)

Es cierto que el manual del C-64 no detalla la forma de representar gráficamente funciones en la pantalla.

Puedes conseguir más información leyendo los artículos de la revista que traten de los gráficos en alta resolución, con la guía de referencia del programador del C-64, o con el manual del SIMON's Basic que acaba de lanzar Microelectrónica y Control.

Soy un VIC-adicto sin remedio desde hace tan sólo ocho meses, ya que fue entonces cuando tuve la oportunidad de tener un VIC-20. Hasta ahora todo había ido bien; los problemas se solucionaban entre el curso de Andrew Colin y vuestra/-nuestra revista (que empecé a recibir en febrero). Pero cuando adquirí el Super-expander (cartucho casi imprescindible) se me presentaron nuevos problemas. Al abrir el folleto adjunto al cartucho, se produjo una lluvia de comandos que después de practicar con ellos logré comprender, pero se me presentaron las siguientes dudas:

1.—¿Cómo funciona y para qué sirve el GRAPHIC-1?

2.—¿Cuáles son las principales aplicaciones del super-expander en video-juegos, compaginando éste con el modo normal de actuar del VIC?

3.—¿Qué pasa cuando a veces se producen fallos en el comando PAINT, que nos lleva a situaciones en teoría típicas del modo GRAPHIC-1?

Asimismo, me gustaría que en sucesivos números fuerais incluyendo más información sobre la forma de lograr interesantes efectos de sonido para juegos y otros programas, y también algún artículo para ampliar conocimientos sobre las funciones de cadena DIMs, y demás misterios, así como de las aplicaciones del plotter VC 1520.

Os felicito por la estupenda labor que realizáis con esta revista.

DAVID PEREZ CORRAL - 15 años
PASSEIG DE SANT JOAN, 171 - 5º 4ª
BARCELONA-37

1.—El modo GRAPHIC 1 del Super-expander, es el modo multicolor en gráficos de mediana resolución. En este modo la pantalla se divide en 160 líneas de 80 puntos por cada línea, y cada punto puede tener su propio color (entre los cuatro seleccionados por el comando COLOR).

2.—La aplicación del cartucho, en juegos es la de facilitarte la programación de tus propios juegos sin tener que recurrir a tantos POKEs como necesitarías si no lo tuvieses.

Los juegos que normalmente puedes encontrar en el comercio, no suelen utilizar el cartucho por estar escritos en código máquina, pero algunos lo necesitan para utilizar los 3K de RAM.

3.—Los "fallos" que mencionas ocurren al utilizar el comando PAINT en el modo de alta resolución, se deben a que el color en este modo se fija por bloques en lugar de hacerlo por puntos.

Apreciados señores:

No ha mucho compréme un COMMODORE-64, a la par que lo hacía un amigo mío. Nuestra primera impresión del aparato fue de cierto desasosiego al comprobar que en el manual adjunto no aparece ni la más mínima explicación acerca de cómo definir en pantalla una recta o una curva en cualquier inclinación.

La guía del comprador de Informática año 1984 publicada por Haymarket, y lo siento si es propaganda, afirma que el COMMODORE-64 posee en su basic instrucciones tan potentes como Point, Line y Circle. Hemos probado todas ellas y el ordenador las ignora por completo.

Por humanidad, ¿cómo dibujar una recta o círculo en alta resolución? Ordenadores "hipotéticamente" menos potentes realizan cualquiera de estas operaciones con una simple línea de programa, y por supuesto sus respectivos manuales dan amplias referencias a esos comandos ¿O es que no se pueden efectuar sin comprar cartuchos adicionales? En ese caso están jugando con la ignorancia y el bolsillo del comprador.

¿Entra ahí el bit-mapping, al que el escueto y superficial manual no hace ninguna referencia?

Creemos que quizás es una decisión precipitada el considerarlo "el mejor ordenador personal del mundo".

Una última súplica. ¿Cómo definir un simple gráfico propio de 8 × 8 pixels? Como de costumbre, aquello que llaman manual, nada aclara. Huelga decir que otros ordenadores "más pequeños" en apariencia al menos, necesitan de pocas instrucciones para representarlos en los respectivos televisores.

Suponemos que más del 99% de los usuarios tendrán estas mismas dudas (sobre todo si poseen solamente el "famoso manual"). Si no fuere así, nos alegramos.

Esperamos no sean reacios a la crítica constructiva y se dignen publicar la carta. Si lo desean pueden reducirla a efectos de espacio, y suprimir lo que creyeran conveniente, pero por favor, publiquen cuanto antes la respuesta.

Les felicito por la revista a la que ya estoy suscrito.

MIGUEL ORTEGA MORALES
C/INFANTE MARTIN, 1, 7º 1ª
TERRASA (BARNA).

El primer asunto que mencionas es problema de la Guia Haymarket y no tenemos nada que ver con ellos ni sabemos qué fuentes de información tienen.

Para dibujar un círculo o una recta con el C-64 tienes varias soluciones:

Hacer un montón de Pokes a la memoria de Pantalla o adquirir un cartucho que amplíe los comandos Basic añadiendo los de gráficos, por ejemplo el Simon's Basic. También puedes esperar a que aparezca publicado algún programa que facilite el trabajo en alta resolución (estamos buscándolo).

El mapeado de pantalla viene explicado en la guía de referencia del programador. Si piensas que es jugar con la ignorancia del comprador... eres libre para opinar como quieras... Todas las casas dicen que el suyo es el mejor ordenador del mundo, es política publicitaria.

En uno de los próximos números publicaremos un programa para definir tu propio juego de caracteres (de 8*8 Pixels), no te podemos prometer cuándo, pero lo haremos.

Estamos intentando cubrir con artículos de la revista las lagunas del "famoso manual" como tú le llamas.



CPU'S EN CANARIAS

Me interesaría saber si existe la posibilidad de conseguir a través de Vds. un catálogo en español sobre el uso del Commodore 64, aparato que yo poseo y que por haberlo comprado en las islas, donde parece ser se importa directamente de Inglaterra, viene con las instrucciones en inglés, lo que, lógicamente, nos ocasiona un grave perjuicio a los usuarios.

Asimismo, en estos aparatos que se pueden adquirir en Canarias, no se puede obtener el sonido a través de la TV con el cable que suministra la casa. Parece ser que es por diferencias determinadas entre el sistema Pal inglés y el que se usa en España. Les agradecería que si Vds. tienen conocimiento de la forma de resolver este problema me lo comuniquen.

Sin más, aprovecho para saludarles y agradecerles de antemano su inestimable ayuda.

FREDDY M. LOPEZ GONZALEZ
PUERTA CARSECO, 47-3º B
SANTA CRUZ DE TENERIFE.

El manual debes pedirlo a Microelectrónica, pero no sé si lo tendrán a la venta. (El del VIC lo anuncian a 800 pesetas). Si tú lo has comprado en Canarias, que es donde vives, el distribuidor que te lo ha vendido es responsable de dártelo debidamente sintonizado. Exígeselo pues es su obligación. Además, este proceso de ajuste es una operación muy simple que lleva unos segundos.

GRAFICAS, DIBUJOS = SUPEREXPANDER

Os envío esta carta pues tengo una duda sobre una instrucción del VIC-20. El problema es que no conozco cuál es la instrucción para poder hacer gráficas, dibujos...

Yo he visto como en el Spectrum utilizan la instrucción PLOT, pero ésta no vale para el VIC-20 y desearía saber cuál es. Espero atentamente recibir respuesta y un pequeño programa de ejemplo, os saluda atentamente.

CARLOS ALBERTO LOPEZ
ROIG, 11 BIS, 4º 3ª
BARCELONA-1

Para disponer de esas instrucciones (Plot, line, circle, draw, etc.) necesitas conectar a tu VIC el Cartucho "Superexpander". Sin él, tendrás que utilizar infinidad de POKES como en algunos de los juegos que aparecen en la revista.

¡MILAGROS, NO!

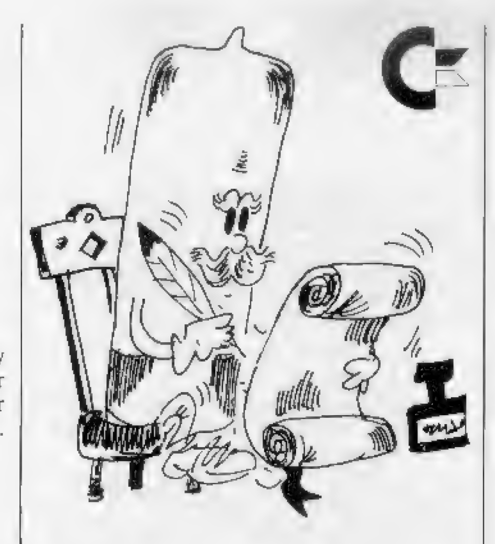
Desearía que me explicarais si hay algún método sencillo para poder utilizar la instrucción/comando AUTO sin poseer el cartucho de "Ayudas al programador". Atentamente,

PERE ESQUERRA (Sin señas)

Los milagros no los hacemos de momento. Métodos "sencillos" sin el cartucho no hay. Podemos intentar estudiar el tema. Lo imposible lo hacemos al momento, con los milagros tardamos un poco más.

POSICIONES EN CODIGO MAQUINA

1.—Tengo unas pequeñas ideas sobre código máquina y entiendo por qué en las posiciones 0, 1 y 2 del VIC-20 hay que introducir los valores que se señalan, pero ¿por qué ha de ser en 0, 1 y 2 y no en otras posiciones posteriores?



2.—Me gustaría saber qué hay que hacer para que al pasar un listado, por ejemplo, en la impresora, las líneas se empiecen a imprimir más hacia la izquierda.

3.—En el VIC-20 hay dos caracteres que se pueden poner desde el teclado, pero que no aparecen en los símbolos frontales de las teclas; salen en pantalla pulsando primero SHIFT y COMM. y después las teclas del número pi y del asterisco junto con la tecla COMMODORE.

FRANCISCO SAEZ SOTO
C/VIRGEN DEL ROSARIO, 28
ALCANTARILLA (MURCIA)

1.—Correcto, lo de los Pokes debes introducirlo en 0, 1 y 2 ya que esa era la función de esas posiciones de memoria. No puedes hacerlo en otras direcciones porque tienen asignadas funciones diferentes.

2.—Tienes dos soluciones para lo del listado:

a) Mover los tractores del papel hacia la derecha lo que te permita el margen que te sobra.

b) Si la impresora fuese programable, alterar el registro que regula el margen (en las instrucciones te lo explicará), pero las impresoras Commodore que yo conozco no pueden programar los márgenes.

3.—Finalmente, tu truco debe referirse a un VIC-20 antiguo, a los que tenemos no les falta ningún carácter.

"SOFT MISTERIOSO"

Aprovecho la ocasión para rogarles incluyan —si es posible— en un próximo número un comentario profundo, sobre todo en su nivel 0, del GRANDMASTER-64 (Nº de catálogo SS080).

FELIX GARCIA GUTIERREZ
CALLE DOCE DE OCTUBRE, 16 INT. 6º B.
VALLADOLID-5

¡Glub! ¡Nos has dejado asombrados! ¿Qué es eso?

Si algún distribuidor nos lo manda, haremos el comentario sobre "esta cosa". Este nº de referencia dinos a qué catálogo de fabricante o distribuidor se refiere.

Repetimos: *Distribuidor y fabricantes:* ¡Necesitamos Catálogos, Información, etc...!

RECHAZO DE CINTAS

Queridos amigos: lo primero que quiero hacer es felicitaros por vuestra revista, que me parece muy completa. A continuación quiero consultaros un problema:

Se trata de mi cassette (Tengo un VIC-20). Tengo varias cintas de juegos y vírgenes; y en ambas hay programas que son aceptados y otros que no (por el ordenador).

Me gustaría saber por qué pasa esto y la solución que podría tener. Gracias.

LEANDRO OLALLA MARTINEZ
EUGENIO SELLES, 5-B (PEDREGALEJO)
MALAGA-17

Si no tienes el datassette de tu VIC, no me extrañan tus pegas... no tiene solución sencilla, solo una: comprar el datassette.

Si lo tienes, puede ser problema de las cintas (cámbialas donde te les vendieron grabadas), o problema del datassette; si es así, debes llevárselo al distribuidor donde lo compraste para que lo comprueben.

"ALERGIA AL CALOR"

Estimados amigos: Empiezo felicitándoos por la magnífica revista que hacéis, de la que soy suscriptor desde el nº 1.

Quería exponeros un problema que tengo con el Super-expander, éste, una vez conectado en el VIC, funciona normalmente, hasta que llevo aproximadamente media hora operando con él, tras la cual, se interrumpe el programa, caso de que esté ejecutando el mismo, o se me borra la pantalla y aparece Ready, si intento listar o volver a ejecutar y no obedece, el ordenador estando bloqueado, caso de desconectar el ordenador, y volverlo a conectar aparece en el monitor toda la pantalla llena de caracteres de colores muy diversos en inversión de color y sigue bloqueado el ordenador, esto persiste hasta otra jornada. Me ocurre siempre, tras la primera media hora. Caso de quitar el super-expander, el ordenador actúa normalmente. Quisiera que me dijerais si es problema del cartucho o del ordenador.

También me gustaría me dijerais quién o dónde, puedo adquirir un programa sobre quinielas para el VIC-20.

Esperando me echéis una mano, recibir las gracias anticipadas de un buen y fiel seguidor.

RAMON PORRERO GARCIA
ESTACION RENFE
CUENCA

El problema seguramente es de tu cartucho, si quieres hacer la prueba, llévatelo a un distribuidor de Commodore que tenga algún VIC-20 conectado todo el día con algún cartucho, y con el VIC "caliente": apágalo, enchufa el cartucho, enciéndelo y prueba. Ya verás cómo el fallo aparece al momento, tu cartucho tiene "alergia al calor".

Si lo pruebas en un VIC de los modernos (que no se calientan por la zona del cartucho), el fallo no aparecerá.

Lleva el cartucho al distribuidor que te lo vendió y él te solucionará tu problema.

Y sobre las quinielas... nosotros también queremos uno... ¡Pero que acierte los 14!

UNIDAD DE DISCOS

Soy un amante de la informática y tengo un problema. Soy poseedor de un VIC-20 con cassette, unidad de disco e impresora. El problema está en que el manual de usuario de la unidad de disco no está muy claro a la hora de abrir, mejor dicho, de trabajar con ficheros.

Yo he probado a manejarlos pero no puedo trabajar con ellos. He trabajado con un ordenador NIXDORF de la serie 8870 con disco y allí es muy fácil abrirlos (cuando digo abrirlos me refiero a trabajar con ellos), y en todo momento se pueden utilizar los registros y sabes qué registros utilizas, pero con esta unidad no lo sabes.

Esperando que haya quedado claro mi problema, se despide vuestro amigo.

P.D.: En el nº 1 de "COMMODORE WORLD" en la página 44 hay un artículo sobre "TECNICAS DE ACCESO DIRECTO. FICHEROS EN DISCO (5)". Me gustaría que si tenéis más articulos sobre ficheros de acceso Directo o relativo me lo enviarais para poder comprender el manejo de la unidad. Gracias.

ANDRES PASCUAL RINCON
FELIX LAFUENTE, 4, 6.º DCHA.
HUESCA

En el número 4 publicamos un artículo sobre la unidad de disco, y en los números 10, 11, 12 y 14 de Club Commodore aparecieron los números anteriores de la serie de artículos "ficheros en disco".

Estamos buscando más información, iremos publicando todo lo que encontremos.

"MAS FALLOS DEL MANUAL"

¡Hola! Me llamo José Luis Tejedor y soy un reciente comprador de un Commodore 64 y lector de su revista.

En el manual del usuario de Commodore 64, en la página 51, en el programa de "juegos de adivinar números", creo que tiene que faltar una línea que fuese CN=0, para que cuando se juegue otra vez de nuevo, los intentos se cuenten a partir de cero y no de los intentos anteriores. Yo tengo puesto lo siguiente: 78 CN=0.

Quisiera saber si lo anterior es correcto y en ese caso si CN=0 hay que colocarlo en la línea 78 o en otra línea.

Les ruego si me podrían decir mi número de socio, pues un amigo mio no se cree que Commodore World tenga más de 1.000 socios.

JOSE LUIS TEJEDOR
C/CARRETAS, 14
MADRID-12

Efectivamente falta la Instrucción CN=0.

Puedes colocarla en cualquier línea a partir de la 2 y antes de la 30, o si prefieres después de la 65 y antes de la 90. Como tú prefieras.

Tu número de socio es el 4.397.

"GRAFICOS EN VIC 20 MAS 16K"

Ruego me indiquen las posiciones a las que se ha de hacer POKE de las datas de los gráficos especiales, a la vez que desco-



nozco si la inicialización POKE 52, 28: POKE 56, 28: CLR: POKE 36869, 255, es válida trabajando con la dicha ampliación de 16 K.

VICENTE GOMEZ FAUBEL
DUQUE DE LIRIA, 70
LLIRIA (VALENCIA)

Esto está explicado en este mismo número con un programa de ejemplo que se llama "El Pozo".

La inicialización sería válida, pero pierdes los 16 K RAM, como si no los tuvieses y cambian las direcciones de los POKES de pantalla.

REGISTROS DE CONTROL

Ahí van algunas preguntas para los expertos en el CBM 64:

1.—¿Cómo fucionan las posiciones de memoria 53272D018 y 53265D011? ¿Se pueden colocar los caracteres programables a partir de la posición 49152? O, mejor aún, ¿podrían colocarse en la RAM que reside "debajo" de la ROM de caracteres? ¿Cómo?

2.—¿Cómo se puede modificar el puntero de lectura de los DATA para que continúe leyendo a partir de una determinada línea de programa? (En algunos sistemas puede usarse la instrucción RESTORE 100 para hacer que el siguiente dato en leerse sea el primero de la línea 100).

3.—Echando un vistazo al BASIC, aparece el comando GO, que por lo tanto no puede usarse como nombre de variable. Sin embargo, no aparece en ningún manual. ¿Para qué sirve?

Y una pregunta para los distribuidores: ¿Tiene COMMODORE la intención de lanzar al mercado otro tipo de unidad de disco más económico? La unidad 1541 me parece demasiado cara, teniendo en cuenta que sólo puede manejar 170 K.

GUILLERMO CAAMAÑO OLIVARES
ARABIAL, 11 - 1º D
GRANADA-3

1.—Esas direcciones pertenecen al Chip de Vídeo del C-64. La explicación sería demasiado larga para esta sección, y la tienes en la guía de referencia del programador del C-64 (páginas 83 y siguientes).

Colocándolos debajo de la ROM tendrás problemas con las unidades de entradas/salidas.

2.—Puedes modificarlo haciendo unos pokes a las posiciones 63, 64, 65 y 66. En uno de los próximos números del misterio de los pokes lo desvelaremos...

3.—Es uno de los que "ni comen ni dejan comer", no tenemos una explicación razonablemente sólida para que no se pueda utilizar, ni como variable ni como comando. Si alguien lo sabe que lo diga...

4.—Lo del precio del 1541 lo dejamos aquí para los distribuidores. A nosotros no nos corresponde entrar ni salir en el tema.

"CAROTENA" DE NUEVO

Ruego me dieran un poco de información sobre el programa "Carotena" remitido en el número de Febrero para que pueda ir en el Commodore C-64.

CARLOS ESTELLES GARGALLO
BALTASAR GRACIAN, 35, 1º 1ª
BARCELONA-16

Ponte en contacto con su autor: Luis Carballo Taboada, C/Cardenal Quevedo, 35, 5º. ORENSE.

De todas formas sus señas aparecen en Carta Blanca del nº 3.

MAS SOBRE LOS POKES

Para los iniciados en el CBM-64 creo que sería importante ampliar información sobre los POKES, porque fuera de los referidos a los Sprites, sonido, caracteres en pantalla, colores y pantallas, no hay muchas fuentes de información (o no sé entenderlas).

JOSE ARTURO GUILLEN VALEN
C/MARINA ESPAÑOLA, EDIFICIO AZABACHE, 14º C.
ZARAGOZA.

En el número 3 hemos comenzado una serie de artículos denominados "El Misterio de los Pokes". Espero que la información que ha aparecido, aparece y aparecerá, te resulte suficiente y puedas entenderla.

NO TENEMOS... NO TENEMOS... NO TENEMOS...

Como componente del antiguo Club Commodore, y suscriptor ahora, del Commodore World, desearía que me enviases, contra reembolso, un esquema de la conexión en paralelo de más de un cartucho al VIC-20, así como información precisa sobre las posibilidades en cálculo y gráficas del cartucho Forth.

Sé que existe un Vic-expander para cuatro cartuchos, y me conformaría con una fotocopia por ambas caras de la chapa del circuito impreso.

Sin más, os saluda, y ójala podáis satisfacer mis deseos.

ANTONIO POLONIO ARMADA
REYES CATOLICOS, 12-114
MONTILLA (Córdoba)

No tenemos el esquema de este expansor, pero si alguien lo tiene publicamos tus señas para que te lo envíe.

Tampoco tenemos fotocopias de la placa del circuito impreso. Información sobre el forth sólo tenemos la lista de comandos que acompaña al cartucho del C-64. Insistimos: Distribuidores, editoriales, etc., ¡Queremos información!

¿64, ORGANO ELECTRONICO, O PIANO?

Os escribo esta carta por dos motivos: El primero es haceros una pregunta sobre el C-64, y la segunda para mandaros una "magia" por si mereciese la pena publicarla.

La pregunta es si es posible utilizar dos teclas a la vez en un programa de alguna forma. Lo pregunto porque he intentado reformar el programa que viene en el libro de instrucciones de teclado de piano, para ver si se podrían hacer acordes.

JOSE LUIS AROSTEGUI PLAZA
ALMONA DEL BOQUERON, 15
GRANADA-1

Los Commodore no pueden detectar cuando se pulsan dos teclas a la vez, con la excepción de SHIFT, CTRL y la tecla COMMODORE. ¡Esto no es un piano!


Las Aventuras de RAMY y ROMO y sus amigos
¡Esto es una memoria?
por ERRAZQUIN
¿OYE RAFS.. ENTONCES RAMY Y ROMO SON DOS MEMORIAS DEL ORDENADOR?
EXACTAMENTE


PUES YA VES.. ME ENCANTARIA SABER COMO FUNCIONA UNA MEMORIA POR DENTRO...
A MI NO


¡PERFECTO! YA TENIA YO GANAS DE UN POCO DE ACCION.. ¡HOP!
BRAVOOOO
QUE YO NO QUIEROOOO
POUF


MEMORIA
ENTRADA SOLO BYTES
MIRA ESA DEBE SER LA ENTRADA
SI.. PERO DICE SOLO BYTES.. Y NI SIQUIERA SABEMOS QUE ES ESO.. ADEMAS HAY SOLDADOS .. VENGA VOLVAMOS..
C
Z


EJEM.. DISCULPEN.. YO OS PUEDO DECIR QUE ES UN BYTE...
SIIIIII QUE BIEN
OH NOOO
A


CADA LETRA, CADA NUMERO, LOS SIMBOLOS.. TALES COMO + o *.. SON CARACTERES.. BUENO PUES A UN CARACTER SE LE PONE, AQUI EN EL ORDENADOR, EL NOMBRE DE BYTE
¡AJA!. TU LLEVAS UNA A.. ES UN CARACTER.. LUEGO TU ERES UN BYTE..
A


CUANDO TECLEAS UN NOMBRE O UNA CIFRA LO QUE LE INTRODUCIS AL ORDENADOR ES UN GRUPO DE BYTES.. Y LOS LLEVA AL PROCESADOR ...... POR CIERTO.. ¿QUEREIS VERLO POR DENTRO?
FANTASTICO ERA JUSTO LO QUE QUERIA
PUES YO NO .. QUIERO VOLVER A CASA
A


TODO EL CONJUNTO CONSTA DE TRES UNIDADES: LA DE CONTROL, LA ARITMETICO-LOGICA Y LA MEMORIA PROPIAMENTE DICHA...
MEMORIA
CONTROL
ARITMETICO-LOGICA
PUES EMPECEMOS POR LA ARITMETICO-LOGICA.. QUE A MI HERMANO SE LE DA MUY MAL LAS MATES..
A


VEIS.. AQUI SE REALIZAN TODOS LOS CALCULOS MATEMATICOS Y OPERACIONES DE COMPARACION TALES COMO ESTO ES MAYOR QUE LO OTRO.. ETC.
VA.
8X
ATENCION UNIDAD ARITMETICO-LOGICA, RESULTADO DE 8 × 3
A


32


¿32?
32
PROCESADO 32


ERROR... UNIDAD EQUIVOCADA ... REPROGRAMAR... ... ERROR
CREO... CREO QUE ES MEJOR QUE NOS VAYAMOS
NOO... NOO.. UN MOMENTO


... QUEDA SUSTITUIDO...
AHHH.. 10 AÑOS EN EL PUESTO Y AHORA ME DEPURAN... BUAAAAA
YO.. YO.. VERA ES QUE ...
VAMOS.. SAL DE AQUI
POM
POM


BIEN.. VISITEMOS AHORA LA MEMORIA CENTRAL.. ¿ VALE?
UNIDAD ARITMETICO-LO
BRIGADA DE REPROGRAMAR


LIBRE
120.301
120.302
129.763
129.764
LIBRE
CELDILLA 130.001
CELDILLA 130.002
LA UNIDAD DE MEMORIA GUARDA TODOS LOS BYTES .. LOS RESULTADOS DE LA UNIDAD ARITMETICO- LOGICA Y LO QUE HA ENTRADO POR TECLADO.. CADA BYTE TIENE UNA HABITACION.. QUE SE LLAMA CELDILLA..


EN CADA CELDILLA HAY UN BYTE.. ENTRAN Y SALEN.. LOS BYTES QUE NO SE USAN SE SACAN DE SU CELDILLA PARA DEJAR PASO A OTROS. ..Y ENTONCES DESAPARECEMOS...
CELDILLA
FUERAA
3
2
POM


OYE BYTE..¿Y QUIEN DECIDE EL NUMERO DE CELDILLA DE CADA UNO DE TUS PRIMOS.. CUANDO HACER LOS CALCULOS O CUALES BYTES HAN DE SUSTITUIRSE POR OTROS?
LA UNIDAD DE CONTROL..HACIA ALLI VAMOS AHORA
ERRAZQUIN

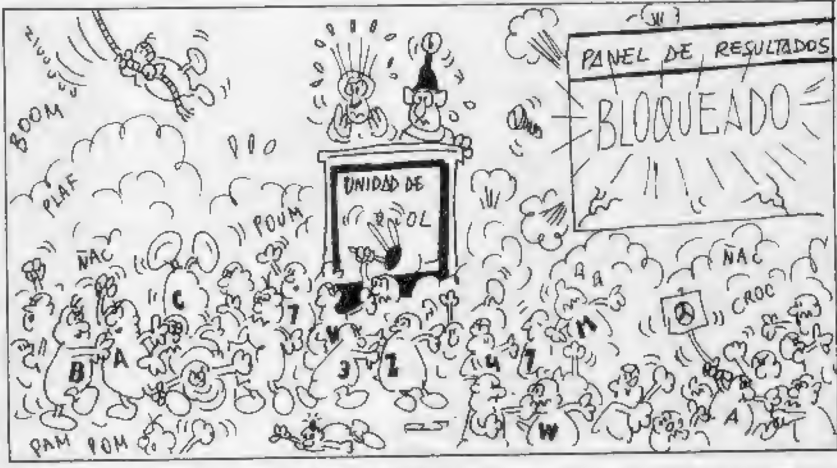

VEIS.. DESDE AQUI SE CONTROLAN LOS NUMEROS DE CELDILLA, LOS RESULTADOS.. LAS PALABRAS QUE DIGITAN EN EL EXTERIOR TODO HA DE HACERSE MUY RAPIDO...
CALCULO YA...
AVANCE LA PALABRA INTRODUCIDA VAMOOS..
SIGUIENTES BYTES.. OCUPEN SUS CELDILLAS .. RAPIDO...
PANEL DE RESULTADOS
8×3 24
VOY A VERLO MAS DE CERCA
PEPE
UNIDAD DE CONTROL
CELDILLA 3838
1982


LIMPIEN ZONA 88 DE MEMORIA.. PASO A UNA NUEVA MASA DE BYTES...
UNIDAD DE CONTROL


¿.. Y VD. QUE BYTE REPRESENTA?
¡HOLA! ENCANTADO NO SOY UN BYTE SOY UN DUENDE Y ME LLAMO RAFSO'RAFS


PERO.. PERO.. ESO NO PUEDE SER.. USTED ES UN BYTE Y..
SE EQUIVOCA SOY UN DUENDE DE ORIGEN IRLANDES
¿PERO QUE PASA ALLI ARRIBA? SE HAN PARADO TODOS LOS PROCESOS..
¿AVISO AL EMPERADOR?


NO PUEDO DEJAR PASAR A NADIE QUE NO SEA UN BYTE PLENAMENTE IDENTIFICADO...
LE REPITO QUE SOY UN DUENDE..
AY
NO EMPUJEN..
DOMINGUERO
A LA COLA
¿QUE COLA? SI HAY SEIS DISTINTAS
¿PERO QUE LE PASA AL DEL CONTROL?
QUE NO ME PISE..
TARUGO


BOOM
PLAF
POUM
ÑAC
UNIDAD DE
PANEL DE RESULTADOS
BLOQUEADO
ÑAC
CROC
PAM POM


SE PUEDE SABER QUE RAYOS OCURRE AQUI..


HOMBRE "EMPE" CUANDO TIEMPO SIN VERNOS ¿COMO ANDA DE SALUD?
SON ELLOS


ATRAPENLOS...


ESTA VISTO QUE NI SIENDO AMABLE NOS PODEMOS ENTENDER.. ¡GROSERO!.. AHORA NOS VAMOS DE AQUI.. ¡HOP!
POUF


PERO.. SI.. SI.. HAN DESAPARECIDO
ERRAZQUIN/84


HAY UNA COSA QUE ME MOLESTA MAS QUE NINGUNA OTRA COSA EN EL MUNDO... ..Y.. ES.. QUE ....


ME BURLEN Y SE ESCAPEN CON UN TRUCO QUE NO TIENE NADA DE LOGICAAA... AHHH.. QUE DESGRACIADO SOY.. BUAAAA... SNIFF
POM
POM
POM



MARKETCLUB

Servicio gratuito para nuestros lectores particulares. Empresas 300 ptas., por línea.

¡De no especificar duración, los anuncios serán publicados durante 1 mes.

MERCADILLO

(Ref. M-27)

• Me gustaría entrar en contacto con gente que tuviera "VIC-20" en Alcorcón para intercambiar programas y experiencias. Mi dirección es: Juan Antonio Barreda García, Avda. Palvoranca, 72, 1º esc. 1º D. ALCORCON (Madrid). Teléf.: (91) 641 07 05.

(Ref. M-28)

• Vendo "VIC-20" (comprado en el 84) "con cassette" y regalo bibliografía, entre ella la suscripción a la revista "Commodore" del año 84. O cambiaría todo ello por cámara fotográfica CANON AE-1, o F1".

Llamar preguntando por Jorge al (94) 462 41 52, de 11 a 2 y de 21 a 23 horas. Jorge Herrán Arnáiz, C/San Ignacio, 3, 3º A. Portugalete (Bizkaia).

(Ref. M-29)

• Vendo el cartucho "Basic Programming (CX2620)" y los dos mandos de teclas (CX50) de Atari. El mejor inicio en la informática. Todo nuevo y por sólo 7.000 ptas. Llamar al 734 11 03. Madrid. David, de 9 a 1 ó por la noche. David Cervigón Fernández, C/Badalona, 102. MADRID-34.

(Ref. M-30)

• Tengo pésimos conocimientos de programación, aunque buenos de astrología, astronomía y mecánica celeste. Me gustaría encontrar un colaborador para construir un ambicioso programa de elaboración de horóscopos astrológicos personales. Yo tengo un VIC-20. Si estáis interesados en esta materia o tenéis alguna información, llamadme. Preguntad por José al 422 66 41 (casa) o también al 332 56 90 (trabajo) de **Barcelona**. Os espero.

(Ref. M-31)

• Me gustaría cambiar juegos para el Commodore 64. El que esté interesado que me mande su lista de juegos y yo le mandaré la mia. Pueden mandar su carta a Juan Carlos Gambín Herrera, P./del Metal nº 1, 3º D. ALICANTE-6. También pueden llamar al teléf.: 280 633. Preguntar por Juan Carlos o Rafael Gambín.

(Ref. M-32)

• Vendo VIC-20 (comprado en enero de 1983) con super expander, cartucho de memoria adicional de 25 kbytes, otro cartucho de ayuda al programador y el monitor de lenguaje de máquina, un expansor de 4 cartuchos y once juegos de Indescomp. Por 60.000 (valor real 98.000). Llamar los días de diario de 7 - 1 por la noche. José Sánchez. Teléf.: (91) 457 01 70.

(Ref. M-33)

• Colegio Nacional de E.G.B. con VIC-20 agradecería recibir toda posible información sobre soft educativo o apropiado para colegios con precios. También acepta cualquier regalo de soft que se le quiera enviar. Colegio Público Antonio Machado. Carlos Arciniega Fernández (Director) - C/Valencia, s/n. SAN SEBASTIAN DE LOS REYES (Madrid). Teléf.: (91) 653 49 35.

(Ref. M-34)

• Deseo ponerme en contacto con programadores aficionados a programas de tipo científico-técnico. Luis Pellicer Lorenzo, C/Conde de Torrecederia, 122, 3º A. VIGO-8. (Pontevedra).

(Ref. M-35)

• ¿Dónde puedo encontrar un ejemplar del libro "MOS 6502 Programming Manual"? Me gustaría echarle un vistazo para ver si es interesante antes de adquirirlo. Luis Suárez Antuña, C/Natividad, 2-4º. ALCALA DE HENARES.

(Ref. M-36)

• Vendo Curso de Introducción al Lenguaje Basic Parte I para el C-64. Precio: 2.000 pesetas. Luis. Teléfono: (91) 241 52 97.

(Ref. M-37)

• Se vende impresora VC-1515 en buen estado. Precio 35.000 ptas. José Soto. CARTAGENA. Teléfono: (968) 51 69 18.

(Ref. M-38)

• Vendo Vic-20 con ampliación de 16K RAM muchos juegos y el cassette y libros incluidos 60.000 pesetas. Teléfono: (91) 446 65 56. Ismael.

(Ref. M-39)

• Vendo Commodore 64 con seis meses de uso, cassette Commodore, 8 programas comerciales de juegos sobre cassette, dos cartuchos de juegos: KIKMAN y WIZARD OF WOR. Cartuchos de FORTH, programas Base de datos y Procesador de textos, ambos en cassette; además un joystick Spectra video y los libros: "Manual de referencia del programador" y "Commodore 64 computing". Dirigirse a Francisco Maya Carrasco, C/ San Fernando, 15. Utrera (Sevilla). Telf: (954) 86 09 49.

(Ref. M-40)

• Vendo lote de cintas para el Commodore 64 compradas en España e Inglaterra y en buenas condiciones. Los títulos son: Skramble, Attack of the mutant camels, 3Dtime treck, 2 cassettes especiales para grabar con el ordenador de alta fiabilidad. Todo por 3.800 pesetas (precio real 6.000 pesetas). Manuel Giner Nadal. Telf: (965) 83 03 08. Hotel "Venta la Chata". CALPE. Alicante.

(Ref. M-41)

• Vendo Commodore 64 y unidad de discos VC-1541 en perfecto estado. Regalo cantidad de programas en disco profesionales; VIC-TREE, Compilador Basic, MACRO ASSEMBLER, HES MONITOR, etc. y cientos de



juegos, muchos inexistentes en España. Todo 155.000 ptas. Francisco Gutiérrez Churtichaga. Telf.: 253 13 40. Madrid. Horas comida y cena.

(Ref. M-42)

• Suministro RTTY para VIC-20 y COMMODORE 64. Cartucho modulador/ demodulador, programa en cassette y manual por 15K. EA3C1W (93) 725 53 80 a partir 22 h.

(Ref. M-43)

• Desearía vender por 40.000 ptas. VIC-20 Datassette E2N (Enero 1984), regalo 18 programas, Guía de referencia del programador y varios libros. También lo cambiaría por pequeño laboratorio fotográfico. J.M. Hervas. Ciudad Real, 10. MONZON. Huesca.

(Ref. M-44)

• Vendo VIC-20: Comprado en diciembre con cassette por 32.000 pesetas; Ramón Baladrón, C/ Novoa Santos, 32-8º. Teléfono: 25 17 29. La Coruña-6.

(Ref. M-45)

• Compraría impresora Seikosha o Commodore. Teléfono: 212 87 69 —Noches— José Luis Domingo.

(Ref. M-46)

• Vendo VIC-20, Cassette C2N, 6 juegos indescomp, 5 cintas con programas, ampliación memoria 16K, 1 cartucho juego, parte 1 del curso de Basic, y 1 libro con programas, todo por 50.000 pesetas. Juan Ventura, C/ Roca, 2-5º 3.ª Sabadell. Teléfono: (93) 725 74 58.

CLUBS

(Ref. C-1)

• Club de amigos del VIC-20 y Commodore 64. Ramón O. Serna Soler. Foto Estudio 2. Plaza de Sombrereros, 2. **PALMA DE MALLORCA.** Teléfono: 21 31 62.

(Ref. C-2)

• Desea información sobre clubs en existencia y gente que quiera formar uno. Agustín Quevedo Velasco. Oña, 55-4º B. **MADRID-34.** Teléf.: 202 94 28.

(Ref. C-3)

• Desea información sobre clubs en existencia y gente que quiera formar uno. Pablo Nistal Alonso. Empecinado, 46, 6º B. **MOSTOLES.** Teléf.: 645 21 70.

(Ref. C-4)

• Club de programación Alaiz, en Pamplona. Está dirigido a estudiantes de B.U.P. y C.O.U. de esta ciudad, y el fin que tiene es la introducción en el mundo de la informática y programación BASIC. No tiene, por lo demás, ningún fin lucrativo. El club propiamente posee equipos COMMODORE 64. VIC-20, y diversos periféficos (impresora, unidad de disco y cassette) CLUB ALAIZ. Pza. Monasterio Santa Gemma, s/n. Teléfonos: 254 480-257 704. **PLAMPLONA.**

(Ref. C-5)

• Club de Commodore de **Albacete**. Fernando Martínez Guerrero. La Roda, 39. 5º D. **ALBACETE.**

(Ref. C-6)

• Círculo Durensan Vigués. **VIGO**. Venezuela, 48. Entrechan. Tels.: (986) 410 683/422 519.

(Ref. C-7)

• Un grupo de usuarios de la Comunidad Autónoma de Murcia intenta crear un Club de Usuarios de VIC-20 y C-64 en esta región, sin fines lucrativos. Interesados llamar a los teléfonos (968) 80 23 64 (Mañanas, preguntar por Paco); (968) 26 72 62 (Tardes, preguntar por Fernando). Francisco Sáez Soto. C/Virgen del Rosario, 28. **ALCANTARILLA.** Murcia.

(Ref. C-8)

• Deseo contactar con interesados en formar un club de microordenadores en Barcelona e intercambiar experiencia Domenec Garrofe. C/Aragón, 575, 4º 2º **BARCELONA-26**. Teléf.: 348 00 39.

(Ref. C-9)

• Interesados en la creación de Club de Usuarios en Girona escribir a "Apartado de Correos nº 580" de **GIRONA.**

(Ref. C-10)

• Xabier Prat i Burdo que vive en c/Rabal Sta. Eulalia s/n de Sta. María Dólo **(BAGES)** Barcelona, quisiera saber si existe algún Club Commodore cerca de Manresa y solicita información sobre todo lo relacionado con Modems.

(Ref. C-11)

• Os rogaría me comunicarais si existe algún Club Commodore en mi ciudad **(GUADALAJARA)** o en **ALCALA DE HENARES** (donde trabajo). También agradecería información acerca accesorios, publicaciones, etc., así como de distribuidores de Madrid que trabajen con el C-64. Miguel Angel Capeña Alonso, C/Tabernero, 16-3º I. Teléf.: (91) 22 67 57.

(Ref. C-12)

• Desearía información sobre la existencia de algún Club Commodore en Somaén o sus cercanías. Carlos Cabrera. Camino de las Eras, s/n. Somaén (Soria).

(Ref. C-13)

• Asociación Manchega de Usuarios de CBM-64 (A.M.U.C.-64) C/Blasco de Garay, 10- ALBACETE. Esta asociación persigue extender el uso racional de la informática, concretándose de momento en el ordenador Commodore CBM-64.

(Ref. C-14)

• Os agradecería que comuniquéis a los lectores de la revista el nacimiento del Club MICRO-ESPLAI, en el que nos reunimos aficionados y profesionales de la electrónica e informática con ganas de divertirnos aprendiendo e intercambiar experiencias.

Estamos en Gran Vía, 682, Pral. (Entre Bruch y Gerona) de Barcelona. Todos los amigos que tengáis más de 15 años y menos de 100 podéis participar de esta experiencia. Y si sóis más jóvenes, venid también, ya nos las ingeniaremos para organizar actividades interesantes.

Para informaros, los viernes de 6 a 8 tarde o los sábados de 9 a 11 de la mañana.

(Ref. C-15)

• Queremos formar un Club Commodore en Barcelona: José Teléfono: 329 86 31. C/ Aurora 19-4º 1.ª Barcelona.

(Ref. C-16)

• Me dirijo a ustedes por un hecho reciente, que es la disolución del único club de usuarios en Catalunya con sede social en ELEKTROCOMPUTER en Barcelona. Quiero decirles que su revista es un medio para crear un nuevo Club, por eso pido que si ustedes lo creen conveniente, publiquen esta carta. Eduardo Macip Fite. C/ Horno de la Villa, nº 4. MONTBLANC. Tarragona.

BOLSA DE TRABAJO

DEMANDA

(Ref. BD-1)

• Realizo procesamiento estadístico de datos. Tfno: 91-273 20 76. Pedro Caminal



Rincón del 700

SOFTWARE PARA EL 700 (6): UTILITARIOS (3)



Copyfile

por Jordi SASTRE

Los utilitarios son herramientas para el programador. El programador trabaja con muchos elementos del ordenador: pantalla, teclado, unidades de disco, impresoras, variables, ficheros, programas,... y un largo etcétera. Pero generalmente la mayoría de los utilitarios se refieren a la unidad de discos en general o a ficheros en particular, que es donde se almacenan los programas y los datos. Así, en meses anteriores hemos visto utilitarios para cambiar el formato de programas, para cambiar números de periférico y para modificar el nombre a los discos. Y continuaremos con utilitarios de discos puesto que no es norma impuesta a la fuerza, sino que se impone por sí misma.

Este mes veremos un copiador de ficheros: el COPYFILE. Antes de que alguien pregunte me apresuro a contestar que sí, el 700 dispone del comando COPY cuyo cometido es precisamente el de copiar ficheros. El problema es que el COPY no puede copiar desde una unidad de discos a otra, sino únicamente entre los drives de la misma unidad o sobre un mismo drive. La razón de ello es que este comando utiliza como "buffer" de trabajo la misma RAM de la unidad de discos, a la que se accede desde ambos drives siempre dentro de la misma unidad. Para copiar entre periféricos diferentes debería haber una función que traspasara esta RAM a través del bus IEEE de una unidad a otra, función que el comando COPY no tiene implementada.

En realidad el problema no sería tal si no fuera por la existencia de los discos fijos Winchester (9060 y 9090). No es imprescindible poder copiar programas o ficheros desde una unidad de diskettes a otra porque siempre se pueden poner los dos discos en la misma unidad, uno en cada drive. Pero no hay manera posible de hacer esto con los discos fijos que, como su propio adjetivo indica, son inamovilbles.

Así que se vio la necesidad de elaborar un programa que permitiera volcar información hacia o desde los discos Winchester. Y por el mismo precio, se obtuvo una segunda ventaja: poder copiar ficheros relativos en formato de DOS 2.5 a formato DOS 2.7, es decir, de formato 8050 a formato 8250. Recordemos que los ficheros relativos tienen una diferencia entre ambos modelos de floppy-disk: los segundos pueden ser más grandes que los primeros gracias a la incorporación del "super side sector", lo que provoca que estos ficheros no sean compatibles entre sí.

De hecho, el programa realiza una función bastante simple: leer bytes de un drive para grabarlos en el otro, pero su enorme ventaja es que esta operación se realiza en código máquina, o sea a gran velocidad. Todo programador con un mínimo de experiencia con Commodore se habrá confeccionado alguna vez un programa para copiar ficheros secuenciales o programas de una unidad a otra. Algo parecido a:

```
10 DOPEN#1, "ORIGEN",DO,U8
20 DOPEN#2, "DESTINO",DO,U9
30 GET#1,A$ : IF ST=2 THEN 60
40 PRIT#2,LEFT$(A$+CHR$(0),1);
50 GOTO 30
60 DCLOSE U8 : DCLOSE U9
```

En la linea 40 puede sorprender la manera de grabar el byte, pero como ya he comentado en un artículo anterior, el leer mediante GET# un byte cuyo valor es cero, la cadena de recepción (en este caso A$) es asignada como cadena nula, por lo que una grabación de ésta cadena sería como no grabar nada: PRINT#2,"";. Efectuando la grabación mediante la fórmula LEFT$(A$+CHR$(0),1) nos aseguramos que va a grabarse el CHR$(0) en caso de que la cadena sea nula, sin afectar para nada cualquier otro valor que pudiera haberse leido en A$ (sólo se graba un carácter).

Este programa Basic resulta lento por dos causas. La primera es que precisamente está hecho en Basic interpretado. La segunda es que cada lectura y cada grabación de cada byte implica la apertura y cierre del bus IEEE, con la consiguiente pérdida de tiempo. COPYFILE soluciona lo primero trabajando en código máquina, y lo segundo operando con toda la RAM disponible en el banco 1 de la CPU como buffer (unos 50 Kbytes).

Por otra parte COPYFILE también permite copiar ficheros relativos, cuya copia en Basic sería un poco más complicada que el programa indicando anteriormente.

Comentemos un poco más profundamente el trabajo de COPYFILE. En adelante, al hablar de "fichero origen" se entiende por el fichero que se desea copiar, mientras que "fichero destino" es el nuevo fichero creado como copia del fichero origen. La copia no puede realizarse en el mismo drive, sino entre drives diferentes o entre periféricos diferentes, porque el fichero destino va a llevar el mismo nombre que el fichero origen.

Pueden ser copiados tres tipos de fichero: el programa, secuencial o relativo. Los dos primeros casos son tratados de la misma manera, salvo en la apertura del fichero (uno es PRG y el otro SEQ). Después sólo tiene que leer bytes del fichero origen guardándolos en el buffer. Si este buffer se llena, lo vacia en el fichero destino y continúa leyendo. Cuando detecta el final del fichero origen, graba lo que le queda en el buffer sobre el fichero destino, y los cierra.

El fichero relativo tiene algo más de complicación, pues interviene el concepto de record. En primer lugar ha de averiguar qué longitud tiene el record. Y lo hace de una manera peculiar. Utilizando un bucle FOR/NEXT intenta abrir el fichero asignando cada vez una longitud diferente (de 1 a 255). Cuando la apertura no emite el error 50,RECORD NOT PRESENT es que ha encontrado la longitud correcta. Este "tanteo" se delata porque durante unos momentos parpadea el led central de la unidad de discos. Una vez abierto el fichero no queda más que leer cada record del fichero origen para grabarlo en el fichero destino.

He descrito el núcleo del programa, el trabajo que se realiza en código máquina. Lógicamente se ha intentado, utilizando el Basic, "adornar" el utilitario para hacerlo más potente y cómodo de utilizar. COPYFILE se compone de dos ficheros: COPYFILE (programa principal en Basic) y COPY.CM (rutina en código máquina). Por tanto, quien desee copiar este utilitario de un disco a otro deberá copiar ambos ficheros.

Para poner en marcha el COPYFILE basta con cargarlo (DLOAD"COPYFILE") y ejecutarlo (RUN). Inmediatamente la rutina COPY.CM es cargada en memoria para su utilización, concretamente a partir de la dirección 1024 ($0400 en hexadecimal) del banco 15. El COPY-



FILE carga el COPY.CM desde el drive 0 del periférico 8. Si se utilizara el programa desde otro drive o periférico, deberá cambiarse la línea 60 de programa.

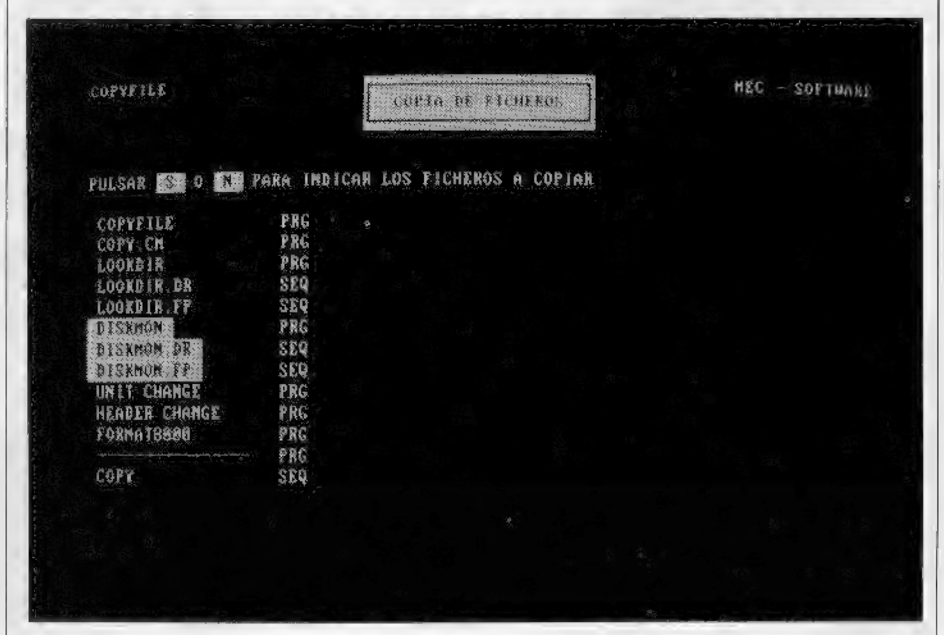
COPYFILE

Una vez en marcha, el programa pide en primer lugar el número de periférico y drive de las unidades de disco que van a utilizarse como origen y destino. He señalado antes que la copia no puede hacerse sobre el mismo drive, sino entre ambos drives de la misma unidad o, por supuesto, entre drives de diferentes unidades de disco.

Una vez comprobada la validez de estos datos pasa a ofrecer las dos opciones posibles de copia:

1. CON VISUALIZACION DE DIRECTORIO.
2. SIN VISUALIZACION (INTRODUCCION DIRECTA)

La primera opción explora el directorio del disco origen visualizando en la pantalla los nombres de los ficheros que va encontrando. El operador debe responder pulsando la tecla S en los ficheros que desea copiar, y N en los que no. Momentáneamente los ficheros a copiar son marcados en negativo en la pantalla, como puede verse en la Figura 1 (los ficheros a copiar son DISKMON, DISKMON.DR y DISKMON.FP). Cuando el programa no encuentra más ficheros en el disco inicia la copia de todos los solicitados. En la pantalla informa en cada momento del fichero que está copiando y del status (DS$) de los que ya ha copiado, por si se hubiera producido algún error de lectura o grabación.



Ejemplo del COPYFILE en funcionamiento. Foto P. Mascats. Diseño Jordi Sastre.

La segunda opción, de introducción directa, es una copia instantánea de ficheros. Es decir, el programa pide un nombre de fichero, el operador lo escribe, inmediatamente este fichero es copiado y el programa vuelve a pedir otro nombre de fichero para copiar. Cuando se pulsa RETURN o ESC sin escribir ningún nombre de fichero, el programa finaliza.

Los nombres de los ficheros a copiar deben escribirse con todas sus letras, es decir, no se aceptan las máscaras de asteriscos ni interrogantes.

COPYFILE seguro que no es el más espectacular de los utilitarios, aunque puede que sea uno de los más útiles. No solo por permitir la copia de ficheros entre dos unidades de discos diferentes, sino también por la ya comentada traducción de los ficheros relativos de formato 8050 a formato 8250, y viceversa. Por otra parte también es más cómodo elegir los ficheros a copiar de entre una lista que el mismo ordenador ofrece a tener que teclear toda una secuencia de comandos COPY, con los consiguientes riesgos de errores humanos y de sintaxis.

NOTICIAS COMMODORE WORLD

Dos navarros, vencedores de un concurso de programación en San Sebastián

Dos jóvenes estudiantes navarros, Camilo Revuelta e Iñaki Sánchez, ambos de 3º de BUP, ganaron el pasado sábado el I Concurso de Programación celebrado en el Colegio Mayor Ayete de San Sebastián. Participaron en él una treintena de estudiantes procedentes del País Vasco y Navarra.



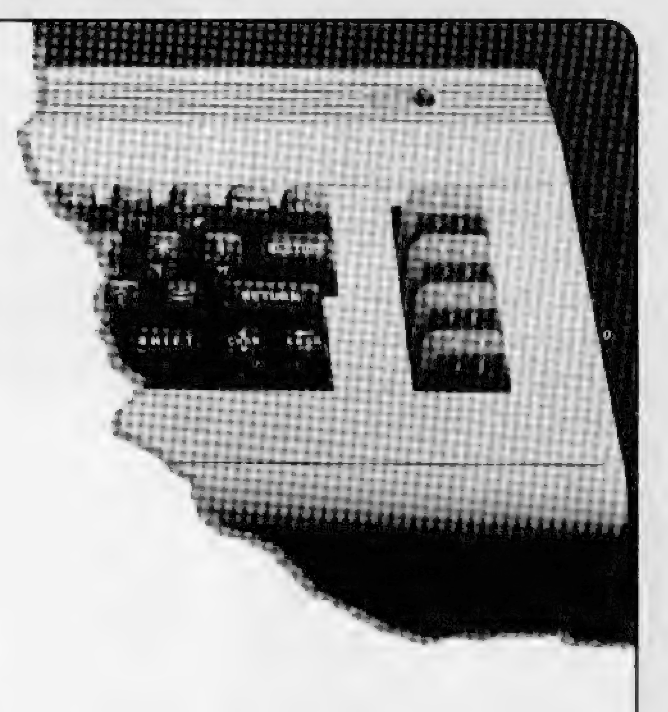
El programa que presentaron los vencedores —al que bautizaron con el nombre de "El barón rojo"— fue elaborado por ellos los días anteriores a la celebración del Concurso, en las instalaciones del Club de Programación Alaiz. Sostuvieron una dura lucha con los demás programas presentados, en especial con el que resultó finalista, presentado por dos estudiantes de Bilbao.

El Concurso de Programas celebrado fue el acto final del Curso Intensivo de Programación que se desarrolló durante la semana de Pascua en el mismo Colegio Mayor Ayete. Allí, además de las clases teóricas y prácticas, se efectuaron visitas a distintos centros de cálculo, y en especial al de la Escuela Técnica Superior de Ingeniería Industrial de San Sebastián, donde algunos profesores explicaron su funcionamiento y mostraron varios programas ya realizados. El último día del curso, un jurado compuesto por varios especialistas y profesionales donostiarras otorgó el primer premio, consistente en un ordenador personal, a los ya mencionados navarros Camilo Revuelta e Ignacio Sánchez. Patrocinaba el curso Comercial Noain, Imagen y Sonido.

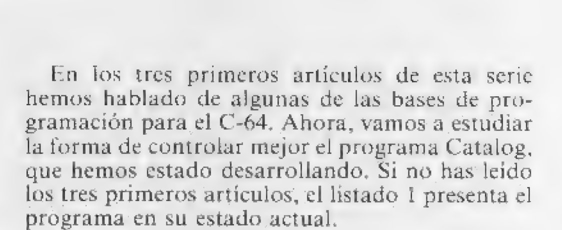


Una Excursión en Basic
Más allá del Manual
Capítulo IV

Por Jeffrey Mills



En este artículo nos despedimos de nuestro viejo amigo, el programa Catalog que hemos visto crecer y desarrollarse a lo largo de estos últimos meses. En el proceso, conoceremos unas sentencias nuevas: Input, If... Then y Restore.



En los tres primeros artículos de esta serie hemos hablado de algunas de las bases de programación para el C-64. Ahora, vamos a estudiar la forma de controlar mejor el programa Catalog, que hemos estado desarrollando. Si no has leído los tres primeros artículos, el listado 1 presenta el programa en su estado actual.

Se inicia la Cuenta de Atrás

Primero, hablaremos del uso de una variable como la forma de contar el número de veces que hemos realizado una operación.

Cuando se asignan variables en un programa, siempre se debe de utilizar la misma variable como contador, para evitar desperdiciar memoria almacenando muchas variables que no se volverán a utilizar. Esta variable-contador puede recibir cualquier nombre, pero para este programa, utilizaremos X. Para que X pueda realizar esta función, usamos esta sentencia de asignación:

LET X=X+1.

A partir de ahora, cada vez que se utilice la sentencia de asignación, será sumando 1 al valor almacenado en el archivo llamado X en la memoria. Esta sentencia Basic realmente significa, "Que el valor almacenado en X sea igual al valor almacenado en X más 1." Utilizaremos un contador en el programa Catalog para limitar el número de entradas a grupos de diez en el listado del catálogo que aparece en pantalla. Esto nos permite disponer de más de diez entradas almacenadas en las sentencias Data sin rebasar los límites de la pantalla. Cada grupo de diez entradas listadas en pantalla será llamado una página. Teclea:

```
58 LET X=0
62 LET X=X+1
80 IF X<10 THEN GOTO 60
```

La línea 58 asegura que la variable-contador contenga un 0 antes de iniciar la cuenta. Esto se llama la inicialización de una variable. Aunque el Basic normalmente inicializa las variables en cero automáticamente, a veces resulta necesario inicializar una variable a un valor que no sea el cero. La inicialización de la variable asegura que contenga el valor necesario. Resulta siempre útil seguir este método.

En el caso de una variable de "strings" (es decir, una variable que contiene caracteres en vez de números), el C-64 inicializa la variable como un nulo, que está completamente vacío; tiene una longitud de cero.

Sin embargo, un blanco ocupa un espacio y tiene una longitud de uno. No confundas un nulo con un blanco. Y ahora, volvamos al contador.

La línea 62 es el contador. Cuando se ejecuta esta línea, el ordenador añade 1 a X, el valor contenido en el archivo.

La línea 80 fue reemplazada por una nueva línea 80, que contiene una sentencia If... Then. Esta línea comprueba para ver si el contador ha llegado a 10, si existen 10 entradas listadas en pantalla.

Si sigue siendo menos de 10, se vuelve a la línea 60 para imprimir otra entrada en el listado del catálogo. Si ha llegado a 10, se pasa directamente al siguiente juego de instrucciones sin imprimir nada más en esta página.

La salida

Si queremos que el ordenador se comunique con nosotros después de haber listado las diez entradas (para preguntarnos si queremos seguir viendo el listado o simplemente terminar), tenemos que estudiar la instrucción Input.

La instrucción Input le indica al C-64 que le haga una pregunta al usuario y que acepte la respuesta desde el teclado. Este constituye un comando potente para la programación de programas tipo pregunta/respuesta, además de poder ser utilizado en juegos controlados desde el teclado. Un nivel de programación más avanzado demostrará que la instrucción Input también puede ser utilizada con ficheros.

Un mensaje le indica al ordenador que te haga una pregunta, o que te pida que pulses una tecla. (Por ejemplo, "INTRODUCE RESPUESTA:" o "PULSA S PARA SI, N PARA NO").

En el caso de nuestro programa. Podemos hacer que el ordenador nos dé el siguiente mensaje:

"¿QUIERES VER LA SIGUIENTE PAGINA (S/N)?"

Dado que la pregunta requiere un carácter o caracteres específicos como respuesta (S o N), hay que presentar las opciones junto con la pregunta; por ejemplo, el "(S/N)" de la instrucción de Input.

La instrucción Input funciona con o sin un mensaje. Si el mensaje resulta ser muy largo, tal vez ocupando más de una línea en la pantalla, se utiliza una sentencia Print para colocar el mensaje en pantalla. Entonces, en la línea siguiente, se utiliza la instrucción Input para conseguir la respuesta desde el teclado.

Si se imprimiera la pregunta como una sentencia distinta, como haríamos en caso de una pregunta larga, y a continuación especificáramos solamente una entrada que nos incluyera la pregunta que se hace, el ordenador contestaría con el signo ?

La instrucción Input puede ser utilizada igual que una sentencia Print; sólo hace falta colocar el mensaje dentro de comillas, y el ordenador hace la pregunta y espera la respuesta dentro de una sola sentencia.

Si se especifica el mensaje como parte de la instrucción Input, el ordenador presenta la pregunta y el signo ? en pantalla.

Es importante recordar que aparece la interrogación (?). El ordenador no sabe si el mensaje incluye una interrogación, especificada entre comillas, y colocaría su propia interrogación al final del mensaje. Por lo tanto, tendríamos dos interrogaciones; no exactamente lo que estábamos buscando.

En cualquier caso, el cursor intermitente aparece detrás del mensaje. De esta forma el ordenador nos indica que está esperando una respuesta.

Utilizaremos el método que incluye la pregunta dentro de la sentencia Input. Tenemos que dejar que se imprima una línea en blanco antes de hacer las preguntas. Tecla:

84 PRINT

```
10 REM *** PROGRAMA/CATALOGO CINTA *
**
20 REM *** ESCRITO POR: TU NOMBRE  *
**
25 POKE53280,12:POKE53281,0:REM ***
COLORES DE PANTALLA ***
30 PRINT"[CLR]"
40 PRINT"[YEL][13SPC]CATALOGO-CINTA"

50 PRINT
52 PRINT"[COMM6]NO.","NOMBRE"
54 PRINT"CINTA","PROGRAMA"
56 PRINT
60 READN,P$
65 IFN=-1THEN GOTO90
70 PRINT"[CYN]"N,P$
80 GOTO60
90 REM *** FIN DEL LAZO ***
9000 DATA 101,JUEGO 1,101,JUEGO 2
9010 DATA 102,JUEGO 3,102,JUEGO 4
9020 DATA 103,JUEGO 5,103,JUEGO 6
9999 DATA -1,FIN DE ARCHIVO (EOF)
```

85 INPUT "QUIERES VER LA PAGINA SIGUIENTE (S/N)"; KB$

Vamos a estudiar el sintaxis (la forma de especificar una sentencia) de la sentencia Input.

Observa la forma en que utilizamos las comillas para especificar la pregunta. Observa también cómo colocamos un punto y coma (;) y un nombre de la variable de "strings" detrás de la pregunta. El ordenador imprime el carácter o caracteres entre las comillas como el mensaje que aparece en pantalla, seguido de una interrogación y el cursor intermitente. La respuesta proporcionada por el usuario se coloca en la variable especificada detrás del punto y coma. En este caso es KB$.

Utilizaremos KB$ como una variable estandard para entradas como esta, que controlan el flujo del programa. Probablemente no se volverá a utilizar el valor KB$, así que no tiene mucho sentido reservar un archivo separado en la memoria para una cosa provisional. KB$ resulta útil dado que implica el uso del teclado; KB$ constituye una entrada provisional desde el teclado. (Key Board=Teclado).

Las entradas no tienen por qué limitarse a los "strings". Se puede pedir una respuesta numérica.

Se pueden solicitar datos múltiples utilizando la misma sentencia Input. Por ejemplo, se puede utilizar un mensaje como "ENTRA MES, DIA, ANO (MM, DD, AA).". Esta entrada podía ir seguida de las variables M, D, y A.

La primera respuesta introducida desde el teclado será introducida en la variable M; la segunda en la D; y la tercera en la A. Todas las respuestas se pueden introducir a la vez tecleando: 4, 15, 84, para contestar al mensaje. Las comas sirven como delimitadores (separadores) y el 4 se coloca en la M, el 15 en la D y el 84 en la A.

También se puede utilizar una respuesta en forma numérica y en forma de "strings" en una sola entrada; por ejemplo. "ENTRA NOMBRE, EDAD," Las variables utilizadas podrían ser NM$ y E. La sentencia Input tendría la siguiente forma:

956 INPUT "ENTRA NOMBRE, EDAD"; NM$, E

La respuesta proporcionada desde el teclado sería:

JOSE LOPEZ, 15

Si la respuesta se entrara de forma incorrecta como:

15, JOSE LOPEZ

NM$ contendría los caracteres 1 y 5 ó 15. Esto funciona dado que el 1 y el 5 también son caracteres además de ser números. Sin embargo, cuando el ordenador intentó introducir los caracteres "JOSE LOPEZ" en la variable numérica E, se produjo un error. El ordenador imprime ?REDO FROM START.

En el caso de nuestro programa Catálogo se ve fácilmente que el ordenador espera una respuesta de S ó N. La sentencia Input se explica ampliamente en las páginas 45-57 del "Manual del Usuario del Commodore 64".

Si nuestra pregunta requiere unas respuestas específicas, como S para si, o N para no, tenemos que incluir una verificación de error para asegurarnos de que se introduzcan solamente las respuestas solicitadas. En el caso de nuestro programa Catalog, queremos asegurarnos de que la respuesta sea un S o un N.

Si la respuesta no es ni S ni N, tenemos que repetir la pregunta y solicitar una res-

```
10 REM *** PROGRAMA/CATALOGO CINTA *
**
20 REM *** ESCRITO POR: TU NOMBRE  *
**
25 POKE53280,12:POKE53281,0:REM ***
COLORES DE PANTALLA ***
30 PRINT"[CLR]"
40 PRINT"[YEL][13SPC]CATALOGO-CINTA"

50 PRINT
52 PRINT"[COMM6]NO.","NOMBRE"
54 PRINT"CINTA","PROGRAMA"
56 PRINT
58 LETX=0
60 READN,P$
62 LETX=X+1
65 IFN=-1THEN GOTO90
70 PRINT"[CYN]"N,P$
80 IF X<10 THEN GOTO 60
84 PRINT
85 INPUT"QUIERES[SPC]VER[SPC]LA[SPC]
PAGINA[SPC]SIGUIENTE[SPC](S/N)";KB$
86 IF KB$<>"S" AND KB$<>"N" THEN GOT
O 85
88 IF KB$="S" THEN GOTO30
89 IF KB$="N" THEN GOTO100
90 RESTORE
92 GOTO60
100 PRINT"[CLR]":END
9000 DATA 101,JUEGO 1,101,JUEGO 2
9010 DATA 102,JUEGO 3,102,JUEGO 4
9020 DATA 103,JUEGO 5,103,JUEGO 6
9030 DATA 104,JUEGO 7,104,JUEGO 8
9040 DATA 105,JUEGO 9,105,JUEGO 10
9050 DATA 106,JUEGO 11,106,JUEGO 12
9060 DATA 107,JUEGO 13,107,JUEGO 14
9070 DATA 108,JUEGO 15,108,JUEGO 16
9080 DATA 109,JUEGO 17,109,JUEGO 18
9999 DATA -1,FIN DE ARCHIVO (EOF)
```

puesta de nuevo. La sentencia If para esto resulta un poco más complicada que las que hemos utilizado hasta ahora, dado que tenemos que comprobar más de una condición (es una S o es una N?).

If... Then

Para comprobar más de una condición, se utiliza la sentencia If... Then igual que se hace para una condición sencilla, pero añadiremos una operación lógica And o una operación Or a la sentencia.

Se utiliza una operación lógica And cuando las dos condiciones que se están comprobando tienen que ser verdad. Se utiliza Or cuando solamente una de las condiciones tiene que ser verdad. Se puede utilizar más de una operación And o una Or en una sentencia If... Then.

Las sentencias If... Then que lleven varios And y Or a veces resultan difíciles de entender, así que, de momento, utilizaremos solamente una And o Or en una sentencia If... Then. Podrás utilizar unas condiciones más complicadas cuando hayas practicado un poco con las operaciones And y Or.

Como esto tiene que ver con el programa Catalog, sólo queremos volver a preguntar si la respuesta es S o N. Para hacer esto se introduce:

86 IF KB$<>"S" and KB$<>"N" THEN GOTO 85.

Recuerda que si se está comprobando una variable de "strings" hay que especificar lo que se está comprobando entre comillas. Los datos especificados entre comillas reciben el nombre de constante o literal. Si la variable es numérica, se pueden seguir utilizando los literales; serán simples números en vez de "strings" y no serán especificados entre comillas.

Para este tipo de comprobación, no tenemos que utilizar los literales para realizar una comparación; se puede utilizar una variable. Sin embargo, para esta aplicación específica, sabemos que la respuesta de la pregunta tiene que ser solamente S o N, por lo tanto se usan literales.

En el caso de esta comprobación, igual que en el caso de la sentencia If... Then en la línea 80, si la condición especificada detrás de la sentencia If es verdad, el Programa realizará lo que se haya especificado detrás de la sentencia Then en la misma línea. Si la condición especificada detrás de la sentencia If NO es verdad, el Programa continuará a la siguiente línea e ignorará la sentencia detrás de la sentencia Then.

Una vez que sepa el ordenador que la respuesta es una letra correcta, se le indica lo que hay que hacer para ver otra página o simplemente terminar. Si se contesta Si —es decir, que el usuario quiere ver otra página— hay que borrar la pantalla y representar la siguiente página del listado.

Dado que queremos imprimir el encabezamiento, tenemos que volver a enviar el programa a la línea 30. Se borrará la pantalla, se imprimirán los encabezamientos y las diez entradas siguientes de las sentencias Data serán leídas e imprimidas.

Ten en cuenta que, al incluir la línea 65, hemos tenido en cuenta, que podemos quedarnos sin dados para imprimir.

Si se contesta No —que no queremos ver más— podemos saltar a una sentencia al final del programa. Quedaría mejor borrar la pantalla al terminar la operación.

Para llevar a cabo estas dos comprobaciones, se teclean estas líneas:

88 IF KB$="S" THEN GOTO 30
89 IF KB$="N" THEN GOTO 100
100 PRINT "[Shift-CLR/HOME]": END.

Antes, la línea 89 había apuntado a la línea 100, y esta línea todavía no existía, y la hemos tenido que añadir. Contiene el comando Print para borrar la pantalla, y pone fin a la ejecución del programa.

Comando Restore

Sería muy útil poder repasar nuestra lista de juegos simplemente contestando S cuando el ordenador pregunta si queremos ver la siguiente página.

De esta forma, si la última página no contuviera diez entradas, nos sería útil poderle decir al ordenador que volviera a iniciar la lista para llenar el resto de la última página.

El comando Restore hace que todo esto sea posible. Devuelve el puntero al princi-

pio de las sentencias Data y permite que el programa empiece a leer desde el principio con la siguiente sentencia Read. La sentencia Restore, en sí, no lee nada. Su función es la de desplazar el puntero en la memoria para que el usuario pueda tener acceso a todas las sentencias Data de nuevo.

Se habla del comando Restore en la página 125 del Manual del Usuario del Commodore 64.

¿Dónde utilizaremos el comando Restore en nuestro programa? Si estuviéramos haciendo este listado a mano, en qué momento tendríamos que volver a hacerlo. Volveríamos al principio de la lista al encontrar, a través de la comprobación en la línea 65, que no nos quedan datos para leer al final de las sentencias Data.

Vamos a añadir un comando Restore a la línea 90 en vez del comentario que lleva, dado que queremos terminar el bucle solamente al contestar N, es decir que no queremos ver la siguiente página.

Vamos a colocar un GOTO detrás del comando Restore para obligar la continuación del listado hasta dar la respuesta N. Esto se llama un GOTO incondicional.

Se emplea el término incondicional si siempre se realiza la misma operación al llegar a una sentencia. Si existiese un motivo para hacer algo o un motivo para no hacerlo (como en la línea 65), se dice que el GOTO es condicional.

El GOTO de la línea 65 es condicional y el GOTO de la línea 90 es incondicional.

Vamos a realizar las siguientes modificaciones al programa:

```
90 RESTORE
90 GOTO 60
```

Hay que volver a la línea 60 dado que el programa tiene que recoger datos de la lista de las sentencias Data. Vamos a hacerlo paso a paso para comprobar si esto es correcto.

En la línea 60 los encabezamientos aparecen en pantalla, y posiblemente hay más datos. Después de que se hayan leído los datos siguientes, la línea 65 comprueba para ver si coinciden con —1, End Of File (EOF). Si estos datos finales son falsos, se pasa a la línea 90.

La línea 90 desplaza el puntero en las sentencias Data hasta el principio. ¡Ahora viene lo más difícil! Si se hubiera vuelto a la línea 70 en vez de a la línea 60, se imprimiría —1 para la cinta y EOF para el nombre del programa. Hay que volver atrás para leer los próximos datos a ser imprimidos a partir del principio de la lista.

Vamos a añadir unas cuantas sentencias Data más al programa, para asegurarnos de que haya más de diez grupos de datos del listado.

Como siempre, con un programa nuevo o en desarrollo, es importante salvar el programa en disco o en cinta antes de que se comprueba. Si existe un error que nos obliga a apagar el ordenador, no habrá ningún problema dado que se carga el programa y se corrige el error.

Ahora, ejecuta el programa para ver lo bien que el C-64 sigue nuestras instrucciones (el listado 2 presenta el programa Catalog terminado).

NOTICIAS COMMODORE

IMPORTANTE INICIATIVA EN EXTREMADURA

El Basic va de veraneo

El Centro piloto "Guadiana" es un colegio público de EGB en el que se está introduciendo la informática a partir del BASIC, utilizando 5 VIC-20 comprados con las 200.000 pesetas otorgadas por el Ministerio al centro para fines educativos.

Van a realizar unas colonias de verano que pueden ser de interés a algún lector de la revista. Estas colonias se realizaran durante el mes de julio y además de contar con las actividades tradicionales de unas colonias de vacaciones como acampadas, deportes (incluido natación), excursiones, convivencias, juegos de interior, etc. han programado dos actividades que han tenido gran aceptación: un curso de BASIC y actividades de granja. Estas colonias tendrán lugar en una granja-escuela que cuenta con instalaciones especialmente indicadas para este tipo de actividades (piscina, campos de deporte, alojamiento, salones, etc), y un personal titulado que se encargará de actividades de granja apropiadas para una colonia de vacaciones (apicultura, injertos, etc.).

En cuanto al curso de BASIC, será impartido por profesores que tienen experiencia en enseñar BASIC a niños de EGB, y lo hacen de una forma muy amena y participativa. Son cursos eminentemente prácticos —un ordenador por cada tres niños en las clases de lenguaje y un ordenador para cada dos niños en las prácticas.

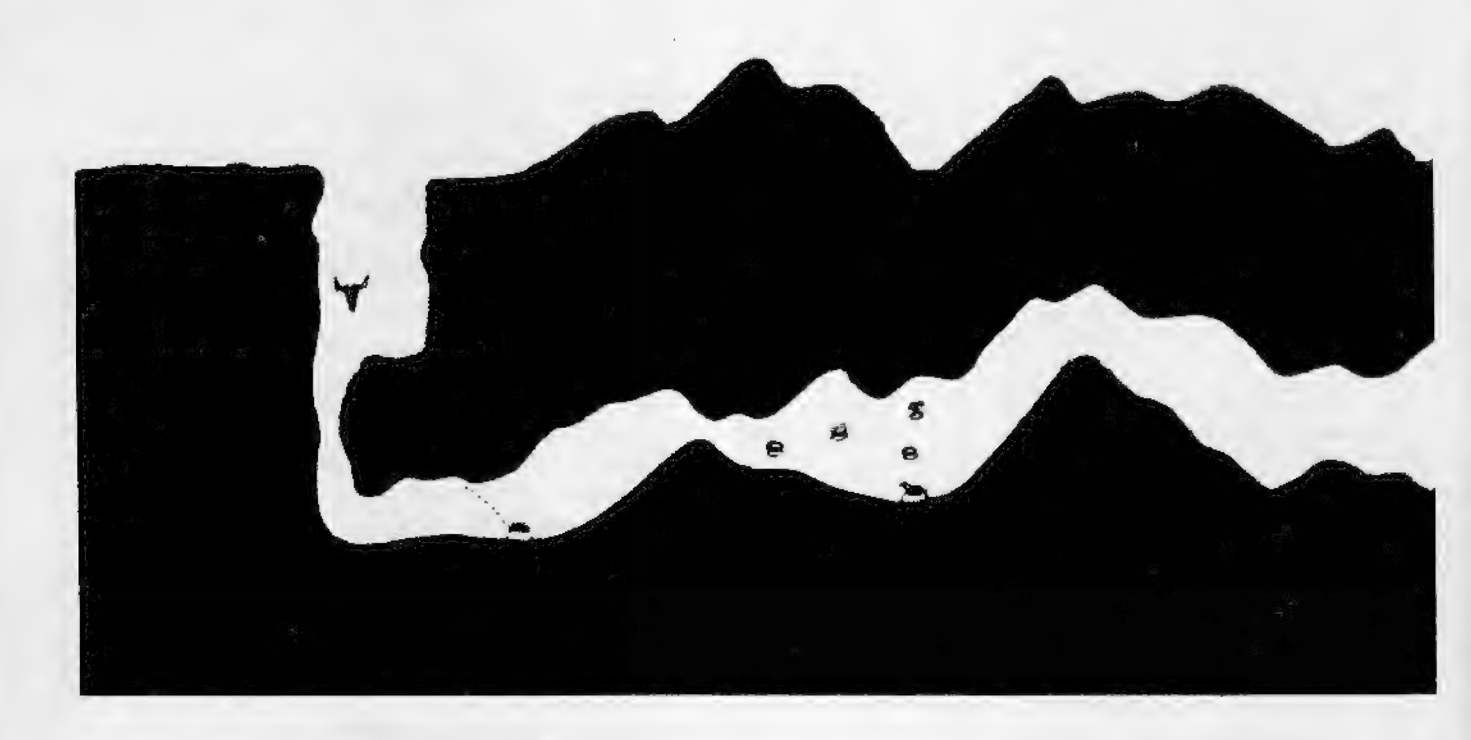
Dado el gran esfuerzo que ha supuesto organizar estos cursos, no se van a limitar a los alumnos del centro sino que se ha hecho una convocatoria abierta (sólo se ha hecho pública en Extremadura). Los cursos, organizados con la colaboración de la Caja de Ahorros de Badajoz, tienen una duración de dos semanas cada uno (del 2 al 14 y del 16 al 28 de julio) y cuestan 19.300 pesetas por curso, todo incluido. El primer curso es de iniciación y el segundo tendrá dos niveles: de iniciación y de perfeccionamiento. El precio del mes completo es de 33.000 pesetas.

Si algún lector de "Commodore World" está interesado, puede solicitar más información a la siguiente dirección:

Señor Director del Centro Piloto "Guadiana"
Avda. República Dominicana, s/n.
Badajoz
Teléf.: 924-250614

Entiende mejor tu ordenador con este divertido juego

Los Peligros del Pozo



Este programa te llevará en un viaje inolvidable hacia las profundidades subterráneas, donde reside una raza de extraterrestres empeñados en cazarte.

VIC-20
C-64

El programa "Los Peligros del Pozo" presenta una persecución animada a través de las profundidades subterráneas de un planeta que tú elijas. Al viajar a toda velocidad en tu nave espacial a través de un pozo muy extraño, tienes que intentar sobrevivir al ataque de unos extraterrestres agresivos. Tú utilizas el "joystick" para trasladar la nave de un lado para otro mientras que el pozo de la mina y los extraterrestres pasan volando.

Tú puedes elegir el nivel de dificultad, la velocidad, la anchura del pozo y el número de extraterrestres. ¿Pero de dónde han salido todos estos enemigos y naves especiales?

La mayoría de los juegos escritos para los Commodore aprovechan al máximo su variado juego de caracteres. ¿Pero qué sucede cuando necesitas algo que no está disponible, como una nave espacial o un extraterrestre, repugnante, por ejemplo?

De ROM a RAM

Afortunadamente, los Commodore te permiten volver a definir completamente el juego de caracteres solamente por la cantidad de memoria disponible. Cuando tú pulsas una tecla, el ordenador busca información

en la ROM (Memoria de sólo Lectura) sobre la manera de formar el carácter que será representado.

Tú seguramente sabrás que no existe un método para escribir en la ROM, y por lo tanto se supone que no hay un modo para intercambiar la información de caracteres. Hasta allí es correcto. Sin embargo, existe una forma de indicarle al micro que no busque los datos de los caracteres en la ROM, sino que verifique las direcciones de la ROM que tú elijas.

En vez de dirigirse directamente a la ROM, el ordenador normalmente busca en una dirección de la RAM (32869 en el VIC) para saber dónde se encuentra la tabla de caracteres. Cuando se enciende el ordenador, la ROM carga el valor de 240 en esta dirección, que apunta las direcciones correctas de la ROM en busca de la información de caracteres. El ordenador siempre se dirige al mismo lugar para buscar los datos, a no ser que otro número se introduzca en esa dirección mediante un "Poke".

De hecho, este valor puede ser modificado a través del teclado. Si se pulsa la tecla Commodore y la tecla "shift" a la vez, el ordenador pasará de la modalidad de mayúsculas y gráficos a la de mayúsculas y minúsculas. Entonces la dirección 32869 (en el VIC) contendrá 242 en vez de 240. Esto también se puede realizar a través de la modalidad de comandos. Inténtalo.

Este juego puede resultar tan sencillo o tan complicado como tú quieras. Pero se garantiza que el nivel de dificultad más alto resulta imposible.

¿Te haces a la idea? También es posible introducir otro número en dicho registro, y hacer que no busque en la ROM, sino en la RAM, donde tú has situado la información sobre el juego de caracteres que tú has elegido.

Pero primero tienes que proteger la RAM de la interferencia del programa bajando el final de la memoria. Esta operación se realiza en las líneas 60-70 (Dado que estos punteros han sido modificados, no es posible utilizar el programa "Los Peligros del Pozo" con un VIC-20 con una ampliación de memoria).

El procedimiento normal para producir unos juegos de caracteres definidos por el usuario es copiar primero toda la información existente sobre los caracteres en la nueva dirección de

Listado VIC-20

```
10 REM ***************
20 REM *             *
30 REM *   EL POZO   *
40 REM *             *
50 REM ***************
60 POKE52,28
70 POKE56,28
80 DIMS$(18)
90 FORN=1TO18
100 S$(N)=""
110 FORN2=1 TO N
120 S$(N)=S$(N)+">"
130 NEXTN2
140 NEXTN
150 PRINT"[CLR]"
160 GOSUB850
170 POKE36869,255
180 FORI=7168 TO 7679
190 POKEI,PEEK(I+25600)
200 NEXTI
210 FORN=7168 TO 7175
220 READH
230 POKEN,H
240 NEXTN
250 FORN=7432 TO 7439
260 READH
270 POKEN,H
280 NEXTN
290 FORN=7664 TO 7671
300 READH
310 POKEN,H
320 NEXTN
330 POKE36879,24
340 VOZ=36876
350 V1=255:V=10
360 DD=37154
370 PA=37137:PB=37152
380 A1=3:FG=1:U1=7680
390 U2=7702:U3=8164
400 POKE36879,14
410 POKE37139,0
420 POKE36878,V
430 POKE VOZ,V1
440 T=1:A=4
450 TI$="000000"
460 DATA226,164,27,40,108,176,175,13
2
470 DATA195,195,68,60,24,24,60,24
480 DATA255,129,189,165,165,189,129,
255
490 T=8:A1=10
500 FORN=1 TO 23
510 PRINTS$(4);SPC(14);S$(3);:PRINT"
"
520 NEXTN
530 GOTO600
540 D=INT(RND(1)*2)+1
550 J=INT(RND(1)*AL)+1
560 IFJ<>2THEN GOTO 620
570 FA=INT(RND(1)*3)+1:FA=FA*D
580 F4=T+INT(W/2)+FA
590 IF F4>T+W OR F4<T GOTO620
600 IF F4=PR GOTO620
610 POKEU3+F4,0
```

Pasa a página siguiente



la RAM. Esta operación proporciona una versión no modificada de los caracteres, en caso de que tú decidas utilizar algunos de los alfanuméricos o gráficos normales. El ordenador no puede ir saltando entre la ROM y la RAM en busca de la información sobre los caracteres, así que resulta necesario porporcionar un juego de caracteres completo para su uso en la RAM.

La operación de copiar se realiza en un bucle For...Next, comenzando en la línea 180 en el VIC y 100 en el C-64. Los valores de los caracteres de la ROM se introducen en la RAM.

Una vez copiados estos valores, se pueden cambiar unos cuantos. Este programa hace uso de unos caracteres no necesarios, como los signos @, ! y >. Estos se encuentran (en las nuevas direcciones de la RAM) en grupos de ocho bytes.

Construcción de los caracteres

Los caracteres se construyen en una matriz de ocho por ocho. Cada bit de un byte representa un pixel en una fila determinada. Cada uno de los ocho bytes representa una fila completa. Podría resultar más fácil ver cómo se construye cada uno de los caracteres especiales. La construcción de la nave especial se presenta en la figura 1.

¿Puedes ver la forma de la nave en dicha matriz? Los bytes para el extraterrestre y para las paredes del pozo de la mina se presentan en las figuras 2 y 3, respectivamente.

Los números decimales que aparecen en la columna de la derecha de cada figura se calculan al convertir los números binarios, y se listan en las sentencias Data del Programa (líneas 460-480 en el VIC, 480-500 en el C-64). Cada línea Data contiene los bytes para una figura distinta. Los números pueden ser modificados para cambiar la forma de los caracteres. El programa realiza este cambio leyendo los datos e introduciéndolos mediante un "Poke" en las direcciones de la RAM.

Una vez inicializado un nuevo juego de caracteres, aparecerá el carácter nuevo en vez del antiguo al pulsar el signo !, @ ó > .

Viene de página anterior

```
620 PR=F4:IFD=2THEND=-1
630 IF T+W+D>18 GOTO660
640 IF T+D<1 GOTO660
650 T=T+D
660 GOSUB720
670 POKEVOZ,0
680 PRINTS$(T);SPC(W);S$(21-W-T);
690 PRINT""
700 POKEVOZ,V1
710 GOTO540
720 GOSUB1110
730 FOR O=1 TO SP:NEXT O
740 POKEU1+A1,32
750 IF S2=-1 THEN A1=A1-1
760 IF S3=-1 THEN A1=A1+1
770 IF A1<1 THEN A1=1
780 IF A1>21 THEN A1=21
790 LF=PEEK(U2+A1):IF LF=32 GOTO820
800 IF LF=33 GOTO820
810 GOSUB1160
820 POKEU2+A1,33
830 POKEU2+A1,33
840 RETURN
850 PRINT"[CLR][2CRSRD][BLU]"
860 PRINTTAB(7)"[RVSON]EL[SPC]POZO
[2CRSRD]
"
870 PRINTTAB(2)"QUE[SPC]VELOCIDAD[SPC]
PARA[CRSRD]"
880 PRINTTAB(2)"EL[SPC]TUNEL?[CRSRD]
"
890 PRINTTAB(2)"[RVSON][RED]1[RVSOFF]
[BLU]=RAPIDO[SPC]A[SPC][RVSON][RED]9
[RVSOFF][BLU]=LENTO"
900 GETA$:IFA$=""THEN900
910 SP=VAL(A$)
920 IFSP<1THEN900
930 SP=SP*2
940 PRINT"[CLR][2CRSRD][SPC]QUE[SPC]
ANCHURA[SPC]DE[SPC]TUNEL[CRSRD]"
950 PRINTTAB(1)"PREFIERES?[CRSRD]"
960 PRINTTAB(1)"[RVSON][RED]1[RVSOFF]
[BLU]=ESTRECHO[SPC]A[SPC][RVSON][RED]
9[RVSOFF][BLU]=ANCHO"
970 GETA$:IFA$=""THEN970
980 W=VAL(A$)
990 IFW<1THEN970
1000 PRINT"[CLR][2CRSRD]CUANTAS[SPC]
VIDAS"
1010 PRINT"[CRSRD][SPC]PARA[SPC]TI"
1020 PRINT"[CRSRD]EN[SPC]EL[SPC]JUEG
O[2CRSRD]"
1030 INPUTMC$:MC=VAL(MC$):IFMC<1 OR
MC>50 THEN1000
1040 PRINT"[CLR][2CRSRD]EXTRATERREST
RES[CRSRD]":PRINT"CUANTOS?[2CRSRD]"
1050 PRINT"[RVSON][RED]1[RVSOFF][BLU]
=MUCHOS[SPC]A[SPC][RVSON][RED]20[RVSOFF]
[BLU]=POCOS[2CRSRD]"
1060 INPUTALIENS$
1070 ALIEN=VAL(ALIEN$)
1080 IFALIEN<1 OR ALIEN>20 THEN1040
1090 ALIEN=ALIEN+1
1100 PRINT"[CLR]":RETURN
1110 POKEDD,127
1120 S3=((PEEK(PB)AND128)=0)
1130 POKEDD,255
1140 P=PEEK(PA):S2=((PAND16)=0)
1150 RETURN
1160 POKEVOZ,0
1170 POKE38422+A1+X1,2
1180 POKEU2+A1+X1,33
1190 POKE36878,15
1200 POKE36877,255
1210 FOR F=1 TO 100
1220 POKE36879,104
1230 POKE36879,14
1240 NEXT F
1250 CR=CR+1
1260 POKE36877,0
1270 POKE36878,8
```

Byte uno:	1 1 0 0 0 0 1 1	195 (decimal)
Byte dos:	1 1 0 0 0 0 1 1	195 (decimal)
Byte tres:	0 1 0 0 0 1 0 0	68 (decimal)
Byte cuatro:	0 0 1 1 1 1 0 0	60 (decimal)
Byte cinco:	0 0 0 1 1 0 0 0	24 (decimal)
Byte seis:	0 0 0 1 1 0 0 0	24 (decimal)
Byte siete:	0 0 1 1 1 1 0 0	60 (decimal)
Byte ocho:	0 0 0 1 1 0 0 0	24 (decimal)

Figura 1. La representación de los bits de la nave espacial en el programa "Los Peligros del Pozo".

El juego

El juego "Los Peligros del Pozo" está basado sobre un principio muy sencillo. Tu nave espacial se introduce mediante un "Poke" en la pantalla en la primera fila, determinándose la posición exacta mediante el movimiento del "joystick", igual que en otros programas. El pozo de la mina y los extraterrestres salen en la parte inferior de la pantalla. Cada pozo o nueva fila de extraterrestres que aparece, hace que la pantalla se

```
1280 IF CR=MC THEN1330
1290 POKEVOZ,V1
1300 POKE36878,V
1310 POKE38422+A1+X1,1
1320 RETURN
1330 FT$=TI$
1340 PRINT"[CLR][2CRSRD][3SPC]FIN[SPC]
DE[SPC]JUEGO.[CRSRD]"
1350 PRINT"[SPC]HAS[SPC]CHOCADO[CRSRD]
"
1360 PRINTCR;"[SPC]VECES.[CRSRD]"
1370 PRINT"[SPC]EN[SPC]";
1380 FT$=MID$(FT$,3,2)+":"+MID$(FT$,
5)
1390 PRINTFT$
1400 PRINT"[2CRSRD][SPC]OTRA[SPC]VEZ
[SPC](S/N)?"
1410 GETA$:IFA$=""THEN1410
1420 IFA$="S"THENPOKE36879,27:POKE36
869,240:RUN
1430 IFA$<>"N"THEN1410
```

Listado C-64

```
10 REM ***************
20 REM *             *
30 REM *   EL POZO   *
40 REM *             *
50 REM ***************
60 PRINT"[CLR]"
70 POKE52,48:POKE56,48:CLR
80 POKE56334,PEEK(56334)AND254
90 POKE1,PEEK(1)AND251
100 FORI=0TO511
110 POKEI+12288,PEEK(I+53248)
120 NEXTI:POKE1,PEEK(1)OR4
130 POKE56334,PEEK(56334)OR1
140 POKE53272,(PEEK(53272)AND240)+12

150 FORN=12288 TO 12288+7
160 READH
170 POKEN,H
180 NEXTN
190 FORN=12552 TO 12552+7
200 READH
210 POKEN,H
220 NEXTN
230 FORN=12784 TO 12784+7
240 READH
250 POKEN,H
260 NEXTN
270 POKE53281,1
280 VOLUME=54296
290 VCE=54273
300 CSCREEN=55296
310 CHAR=1024
320 DF=CSCREEN-CHAR
330 DIMS$(40)
340 FORN=1TO40
350 S$(N)=""
360 FOR N2=1 TO N
370 S$(N)=S$(N)+">"
380 NEXTN2
390 NEXTN
400 PRINT"[CLR]"
410 GOSUB860
420 POKE53281,6
430 V1=255:V=10
440 A1=3:FG=1:U1=1024
450 U2=1064:U3=2023
460 T=1:A=4
470 TI$="000000"
480 DATA226,164,27,40,108,176,175,13
2
490 DATA195,195,68,60,24,24,60,24
500 DATA225,129,189,165,165,189,129
255
510 T=8:A1=10
520 FORN=1 TO 40
530 PRINTS$(4);SPC(14);S$(3);:PRINT"
"
540 NEXT
550 GOTO620
560 D=INT(RND(1)*2)+1
570 J=INT(RND(1)*AL)+1
580 IFJ<>2 THEN GOTO640
590 FA=INT(RND(1)*3)+1:FA=FA*D
600 F4=T+INT(W/2)+FA
610 IFF4>T+W OR F4<T GOTO640
620 IFF4=PR GOTO640
630 POKEU3-40+F4,0:POKEU3-40+DF+F4,1

640 PR=F4:IF D=2 THEN D=-1
650 IFT+W+D>38 GOTO680
660 IFT+D<1GOTO680
670 T=T+D
680 GOSUB730
690 PRINTS$(T);SPC(W);S$(39-W-T);
700 PRINT""
```

desplace hacia arriba, acercándose a tu nave. El programa comprueba para ver si la posición que ocupará tu nave es un espacio. Si no lo es, tu nave ha chocado.

Tanto el desplazamiento del pozo hacia la derecha o la izquierda como las posiciones de los extraterrestres se elige al azar. Sin embargo, tú decides la frecuencia con que aparecen. El juego puede resultar tan sencillo o tan complicado como tú quieras. Pero, cuidado, no elijas el pozo más estrecho, ni la velocidad más rápida.

1 1 1 0 0 0 1 0	226
1 0 1 0 0 1 0 0	164
0 0 0 1 1 0 1 1	27
0 0 1 0 1 0 0 0	40
0 1 1 0 1 1 0 0	108
1 0 1 1 0 0 0 0	176
1 0 1 0 1 1 1 1	175
1 0 0 0 0 1 0 0	132

Figura 2. La representación de los bits del extraterrestre del programa "Los Peligros del Pozo".

1 1 1 1 1 1 1 1	255
1 0 0 0 0 0 0 1	129
1 0 1 1 1 1 0 1	189
1 0 1 0 0 1 0 1	165
1 0 1 0 0 1 0 1	165
1 0 1 1 1 1 0 1	189
1 0 0 0 0 0 0 1	129
1 1 1 1 1 1 1 1	255

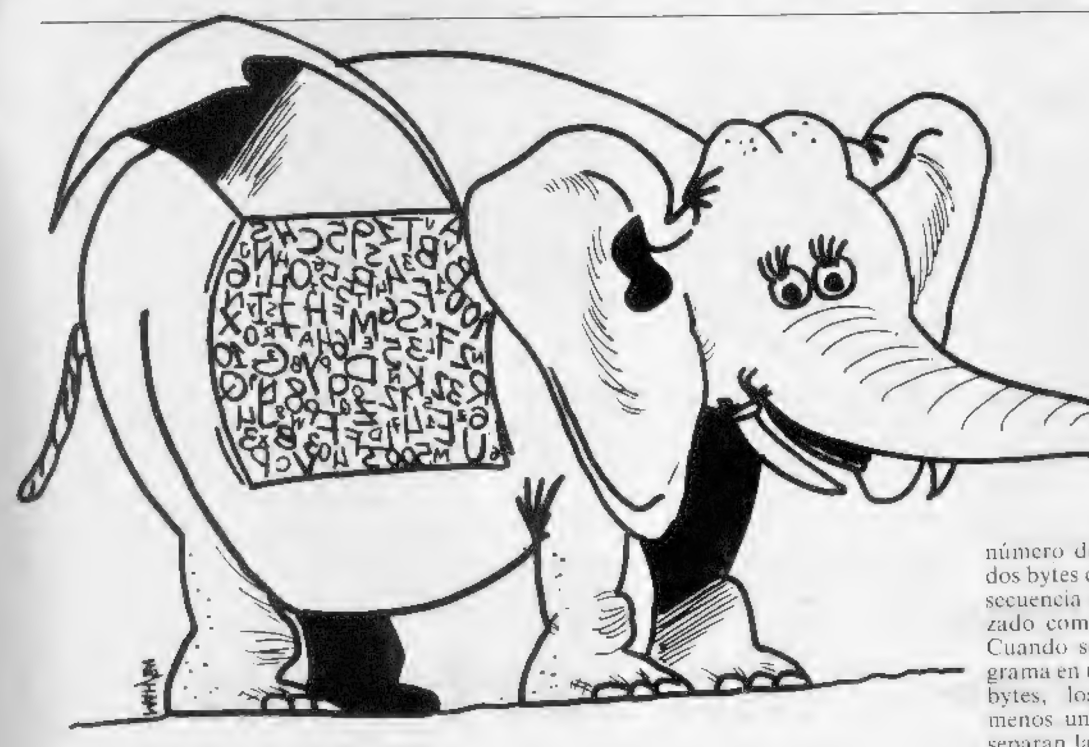
Figura 3. La representación de los bits de las paredes del pozo del programa "Los Peligros del Pozo".

```
710 GOSUB1420
720 GOTO560
730 GOSUB1150
740 FOR O=1 TO SP:NEXT O
750 POKEU1+A1,32
760 IFS2=-1 THENA1=A1+1
770 IFS3=1 THENA1=A1-1
780 IFA1<1THENA1=1
790 IFA1>39THENA1=39
800 LF=PEEK(U2+A1):IF LF=32GOTO830
820 GOSUB1210
830 POKEU1+A1,33
840 POKEU1+A1+DF,1
850 RETURN
860 PRINT"[CLR][2CRSRD][BLU]"
870 PRINTTAB(13)"EL[SPC]POZO[2CRSRD]
"
880 PRINTTAB(8)"QUE[SPC]VELOCIDAD[SPC]
PARA[CRSRD]"
890 PRINTTAB(13)"EL[SPC]POZO?[CRSRD]
"
900 PRINTTAB(6)"[RED]1[RVSOFF][BLU][SPC]
(RAPIDO)[SPC]A[SPC][RED]9[RVSOFF][BLU]
[SPC](LENTO)"
910 GETA$:IFA$=""THEN910
920 SP=VAL(A$)
930 IFSP<1THEN910
940 SP=SP*2
950 PRINT"[CLR][2CRSRD]"TAB(8)"QUE[SPC]
ANCHURA[SPC]DE[CRSRD]"
960 PRINTTAB(8)"POZO[SPC]PREFIERES?
[CRSRD]
"
970 PRINTTAB(5)"[RED]1[RVSOFF][BLU][SPC]
(ESTRECHO)[SPC]A[SPC][RED]9[RVSOFF][BLU]
[SPC](ANCHO)"
980 GETA$:IFA$=""THEN980
990 W=VAL(A$)
1000 IFW<1THEN980
1010 PRINT"[CLR][2CRSRD]";TAB(8)"CUA
NTAS[SPC]VIDAS"
1020 PRINTTAB(8)"[CRSRD]PARA[SPC]TI"

1030 PRINTTAB(8)"[CRSRD]EN[SPC]EL[SPC]
JUEGO[2CRSRD]"
1040 PRINTTAB(8)"";
1050 INPUTMC$:MC=VAL(MC$):IFMC<1 OR
MC>50 THEN1010
1060 PRINT"[CLR][2CRSRD]"TAB(7)"CUAN
TOS[SPC]EXTRATERRESTRES[SPC]?[CRSRD]
"
```

```
1070 PRINTTAB(8)"[RED]1[RVSOFF][BLU]
[SPC](MUCHOS)[SPC]A[SPC][RED]20[RVSOFF]
[BLU][SPC](POCOS)[2CRSRD]"
1080 PRINTTAB(8)"";
1090 INPUTALIENS$
1100 ALIEN=VAL(ALIENS$)
1110 IFALIEN<1 OR ALIEN>20 THEN1060
1120 ALIEN=ALIEN+1
1130 PRINT"[CYN]"
1140 RETURN
1150 JV=PEEK(56320)
1160 JV=15-(JVAND15)
1170 IFJV AND4 THEN S3=1:S2=0:GOTO12
00
1180 IFJV AND8 THEN S2=-1:S3=0:GOTO1
200
1190 S2=0:S3=0
1200 RETURN
1210 POKECSCREEN+A1+X1,2
1220 POKEU2+A1+X1,33
1230 POKEVOLUME,15
1240 POKEVCE,40:POKE54276,23
1250 FORF=1 TO 100
1260 NEXT F
1270 POKEVOLUME,0
1280 CR=CR+1
1290 IF CR=MC THEN GOTO1320
1300 POKECSCREEN+A1+X1,1
1310 RETURN
1320 FT$=TI$
1330 PRINT"[CLR][2CRSRD]"TAB(12)"FIN
[SPC]DE[SPC]JUEGO.[CRSRD]"
1340 PRINTTAB(8)"HAS[SPC]CHOCADO[CRSRD]
"
1350 PRINTTAB(8)CR;"[SPC]VECES.[CRSRD]
"
1360 PRINTTAB(8)"[CRSRD]EN[2CRSRD]"
1370 FT$=MID$(FT$,3,2)+":"+MID$(FT$,
5)
1380 PRINTTAB(12)FT$
1390 PRINTTAB(12)"[2CRSRD]OTRA[SPC]V
EZ[SPC](S/N)?"
1400 GETA$:IFA$=""THEN1400
1410 IFA$="S"THENRUN
1415 IFA$<>"N"THEN1400
1417 END
1420 POKEVOLUME,15
1430 POKEVCE,32
1440 POKEVCE,0
1450 POKEVOLUME,0
1460 RETURN
```

Ahorrando memoria en el VIC-20



Aprende algunas técnicas para sacarle el jugo al VIC-20 no ampliado. No sea VICtima de la crisis de memoria.

Si utilizas un C-64, la RAM disponible para que el usuario escriba sus propios programas normalmente es suficiente. Sin embargo, un VIC-20 no ampliado limita seriamente las capacidades del usuario.

Esos trucos podrían ser de ayuda. Al final, el usuario seguramente verá la necesidad de añadir más memoria, pero incluso en ese caso, estas técnicas te permitirán hacer "más con menos".

La técnica más común para salvar espacio es la de no utilizar comentarios en el programa. Esto, sin embargo, dificulta la documentación del programa y si el usuario quiere realizar algún cambio más adelante, podría tardar bastante en recordar la función de cada variable y cómo se ejecuta el programa. Para más seguridad, es mejor tomar apuntes y guardarlos a medida que se escribe el programa.

Otra técnica es la de poner varias sentencias en una línea, haciendo uso de dos puntos para separarlas. Se desperdician cinco bytes en cada línea de programa en Basic, sin tener en cuenta su longitud. Estos consisten en dos bytes para el número de línea (codificado en binario), dos bytes que apuntan la siguiente línea de secuencia en Basic, y un solo byte utilizado como el indicador de fin de línea. Cuando se combinan dos líneas de programa en una sola línea, se ahorran cuatro bytes, los cinco bytes desperdiciados menos un byte para los dos puntos que separan las sentencias.

Pero hay que tener cuidado; no es siempre válido o correcto combinar las líneas del programa. Se podría modificar el flujo del programa o crear parte de una línea del programa que nunca llegaría a ejecutarse. Hay que tener un cuidado especial con las sentencias If... Then... y GOTO. Merece la pena comprobar que la nueva línea de programa sigue realizando lo que se deseaba.

VIC-20

Más trucos para ahorrar espacio

Otro método para ahorrar espacio es el de borrar todos los espacios no necesarios del programa. El Basic de Commodore no requiere espacios de separación dentro de las sentencias del programa. Las palabras clave en Basic se almacenan como signos codificados de un solo byte cuando se introduce una línea de programa mediante el teclado. Cuando el programa se interpreta y se ejecuta, los signos se identifican fácilmente dado que los códigos que poseen son únicos. Los espacios de separación son simplemente ignorados.

Aunque las palabras clave en Basic se almacenan como signos de un solo byte, el resto del texto en una línea del programa se almacena igual que se teclea, un carácter por byte. Por lo tanto, se puede ahorrar más espacio si se utilizan números de

línea más bajos y nombres devariables más cortos. Si se utilizan los números de línea 1, 2 y 3 en vez de 1000, 2000 y 3000, se ahorran tres bytes en cada sentencia GOTO o GOSUB que utilice estas líneas. Además, las variables se pueden aprovechar al máximo, igualándolas a los valores utilizados con más frecuencia. Esto es aplicable especialmente a los "strings". El puntero de las variables de "string" apuntará al texto de la línea del programa Basic donde se define la variable, en vez de utilizar el espacio de variable al final de memoria.

Además, es mejor volver a utilizar las variables, siempre que sea posible, en vez de definir variables nuevas cada vez que hace falta realizar un cálculo. Cuanto menos espacio se utilice para las variables, más espacio estará disponible para el programa.

Se debe de evitar el uso de "arrays" (para permitir almacenar más datos) cuando las variables simples sirven igual, dado que un "array" tiene más desperdicio. Si resulta necesario utilizar un "array", es imprescindible declarar su tamaño y utilizar el elemento cero.

Hay que tener cuidado con el uso de las variables de enteros. Las variables de enteros simples ocupan siete bytes cada una, mientras que los elementos de los "arrays" de las variables ahorran memoria mediante el uso de solamente dos bytes por elemento. Si se utiliza una variable de entero simple como K%, se desperdicia mucho espacio con el uso del signo del porcentaje cada vez que se hace referencia a la variable.

Además, es importante recordar que el Basic fija un valor por defecto para cada variable la primera vez que ésta se encuentra durante la ejecución del programa; los "strings" se fijan en un "string" nulo y las variables numéricas se fijan en cero... No hace falta inicializar las variables a estos valores, dado que el Basic ya lo hace.

Constantes y Comillas

Los datos constantes utilizados por un programa pueden ser introducidos en un "array" mediante lectura a partir de un fichero de datos externo de cinta o disco, en vez de hacerlo a partir de las sentencias de datos dentro del programa. Por otro lado, los datos se pueden utilizar directamente de las sentencias "Data" sin que se coloquen en un "array", y se pueden volver a utilizar todas las veces necesarias mediante el uso del comando "Restore".

También se puede ahorrar espacio omitiendo las comillas que contienen los valores de los elementos "string" en las sentencias "Data". Las comillas sólo son necesarias si existen espacios o caracteres especiales como gráficos, controles del cursor, comas, dos puntos, etc., dentro de las sentencias "Data".

También se pueden omitir las comillas de cierre en cualquier sentencia Print si ésta no va seguida por otra unidad de información que se tiene que imprimir dentro de la misma sentencia, o por dos puntos y otra sentencia Basic. El Basic añadirá automáticamente las comillas de cierre e imprimirá la línea.

Siempre que sea posible se debe de saltar la puntuación dentro de las sentencias Print que contengan varias unidades de información. Si se imprimen las variables separadas por un texto, como, por ejemplo.

```
100 PRINT "LENGHT=";X;"
    WIDTH=";Y;" AREA=";Z,
```

resulta más fácil utilizar:

```
100 PRINT "LENGHT="X"
    WIDTH="Y" AREA="Z
```

No hace falta utilizar los puntos y comas de separación, dado que el intérprete Basic da por hecho que existe un punto y coma de separación entre las unidades de información.

Se utilizan las funciones TAB y SPC para evitar el uso de espacios adicionales en las sentencias Print. Hay que tener en cuenta que estas funciones desplazan el cursor a la derecha el número de espacios necesarios. Estas funciones no se pueden utilizar para borrar algo de la pantalla, dado que un carácter de espacio no se imprime.

Además, las subrutinas siempre deben utilizarse para realizar las funciones normales que se necesitan en varios puntos determinados del programa. Si se necesita del usuario de si/no en varios puntos a lo largo del programa, esta función debe ser realizada por una subrutina, la cual será llamada todas las veces necesarias.

Si existen bucles anidados de For... Next que tengan un punto común de salida, las siguientes sentencias Next deben ser combinadas:

```
100  FOR X=1 TO 10
110  FOR Y=10 TO 100
120  FOR Z=2 TO 4
*
*
*
200 NEXT Z,Y,X
```

Hay que tener cuidado de listar las variables en orden correcto, y no olvidar que el bucle interior terminará primero.

Finalmente, hay que evitar el uso de los paréntesis cuando no son realmente necesarios para la interpretación correcta de la sentencia. Es imprescindible recordar el orden de prioridades de las operaciones demostradas en los manuales.

Espero que esta información resulte útil para que tus programas quepan en el espacio disponible. En general, la mayoría de las técnicas para ahorrar espacio también proporcionarán un tiempo de ejecución más rápido.

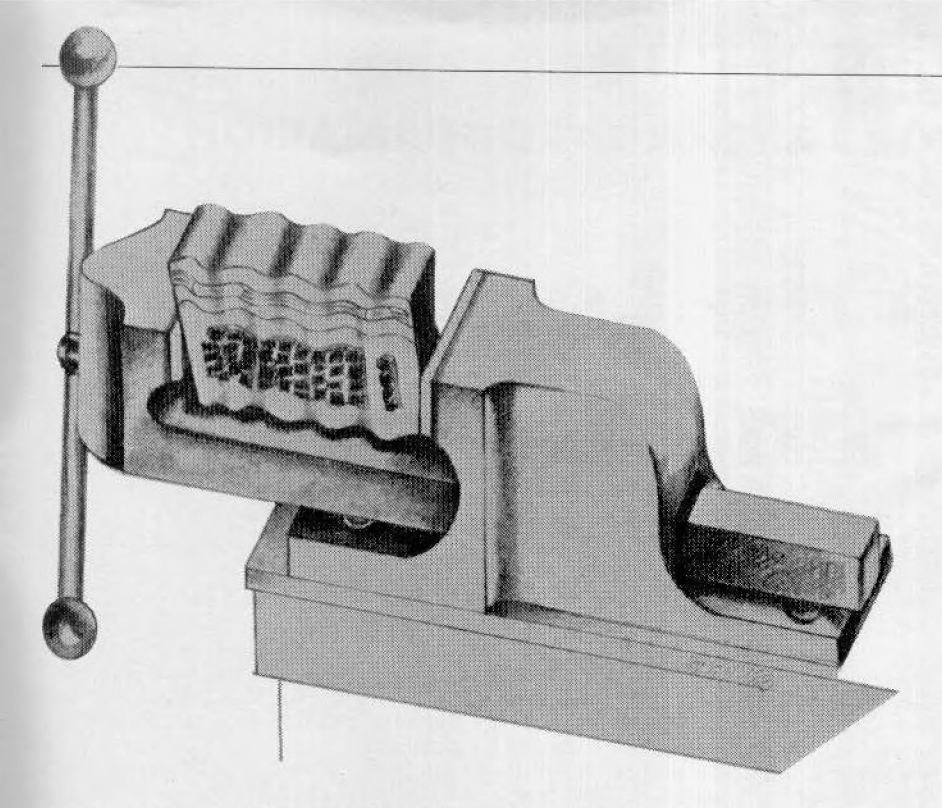
Experto en ahorrar espacio

Y ahora... Es la hora de ahorrar tiempo.

Después de aprender a ahorrar espacio cuando escribes tus programas en Basic, ha llegado el momento de aprender a ahorrar tiempo también. Sin embargo, hay que tener en cuenta que la mayoría de las técnicas utilizadas para ahorrar tiempo podrían suponer un incremento del tamaño del programa.

Para empezar, se deben de utilizar las variables en vez de las constantes, dado que se recupera el valor de una variable simple o de "array" de las tablas de variables más rápido que realizar la operación de convertir una constante de una sentencia de programa en su representación interna de punto flotante. Por ejemplo, si se utiliza el número 256 varias veces a lo largo del programa, hacemos P=256 y que se utilice la variable P cuando se hubiera usado 256.

Las definiciones de las variables se tienen que ordenar con cuidado. Las variables se definen en la tabla interna de variables en el orden en que se encuentren durante la ejecución del programa. Cuando el programa se refiere a una variable, el Basic tiene que barrer la tabla de variables desde el principio para encontrar el valor deseado. Dado



que las variables que se hayan definido primero son las primeras que se localizan, será más útil si se definen primero las que se utilizan más. Incluso podría ser una ventaja definir ciertas variables con valores de prueba al principio del programa para que entren en la tabla de variables en un orden específico.

Es mejor definir las variables simples antes de utilizar un "array" largo. Cuando se define una variable simple, el Basic debe desplazar los "arrays" existentes en la memoria para hacer espacio para la entrada de la nueva variable simple, y de acuerdo con esto debe modificar todos los punteros de "arrays" relacionados con dicho "array".

Es aconsejable colocar las subrutinas más utilizadas cerca del principio del programa y utilizar una instrucción simple de GOTO para saltar las subrutinas cuando empieza el programa. El Basic normalmente busca a partir del principio del programa para encontrar la línea de la subrutina, así que si se colocan las subrutinas cerca del final del programa se pierde mucho tiempo, sobre todo si la subrutina se utiliza para leer o escribir los ficheros de datos.

En realidad, el Basic no siempre tiene que barrer el programa a partir del principio del programa para encontrar la línea de la subrutina. Si el número de línea llamado es mayor por 256 por lo menos que la línea que llama la subrutina, el Basic empezará a barrer a partir de la línea actual. Sin embargo, de esta forma también se puede perder el tiempo, dado que la subrutina no está siempre cerca de la línea que hace la llamada.

Next-For

Examina el contenido y la estructura de cada bucle For-Next, el cual es posiblemente el proceso que más tiempo pierde si no se utiliza como es debido. Las sentencias que no necesitan repetirse para cada repetición del bucle deben quedarse fuera del bucle. Si un valor tiene que calcularse o un valor del "substring" se utiliza una y otra vez dentro del bucle, se debe de calcular el valor una vez y definirlo como otra variable que puede ser utilizada según sea necesario.

Ya que estamos hablando de los bucles For-Next, es importante acordarse de no utilizar las sentencias Next con las variables de índice. La sentencia Next resulta un poco más rápida que Next X dado que no se realiza ninguna verificación para saber si la variable utilizada realmente es la misma que la de la sentencia For más reciente.

Otra técnica para ahorrar tiempo es la de combinar operaciones cuando sea posible dentro del flujo normal del programa. Si el programa va a introducir datos mediante la lectura en un "array" y a continuación los va a clasificar, resulta útil colocar la entrada en la posición correcta dentro del "array" y desplazar las entradas restantes de acuerdo con esto a medida que se lee cada unidad de información.

Este método resulta especialmente útil cuando los datos se introducen manualmente desde el teclado. A lo mejor se aprecia una pequeña pausa entre una entrada y la siguiente, pero más adelante, cuando haya que clasificar una matriz entera una pausa más larga podría resultar molesta.

Como dije antes, los programas más rápidos no son siempre los más cortos; normalmente se puede llegar a un compromiso. Resulta mucho más rápido imprimir la letra A 40 veces con una sola sentencia Print "AA... AA" que tener que utilizar un bucle For... Next como el que sigue:

```
For X=1 TO 40; Print "A";: Next X
```

Por otro lado, el bucle For... Next ahorra bastante espacio de memoria comparado con la sencilla sentencia Print. Todo depende de lo que quieres conseguir y lo que tú consideras más importante: ahorrar tiempo o espacio. A veces estas dos cosas pueden resultar totalmente opuestas. ■

EXCURSION POR LA MICROPROGRAMACION

Cursillo de Lenguaje Máquina —III—

Por Diego ROMERO

Al igual que en BASIC podemos colocar los datos de un programa en el propio programa con A=24 o con sentencias DATA en una tabla y hacer READ A ó con un INPUT A y dándole el dato desde el teclado, en código máquina podemos colocar los datos en distintos sitios, y para que la CPU pueda acceder a ellos, está dotada de distintos "Modos de Direccionamiento", vamos a estudiarlos un poco y más adelante volveremos a verlos más detenidamente.

El modo de direccionamiento es la manera en que cada instrucción de un programa de código máquina accede a los datos sobre los que tiene que operar, el más sencillo es el inmediato, veamos este modo y su equivalente en basic para facilitar la comprensión.

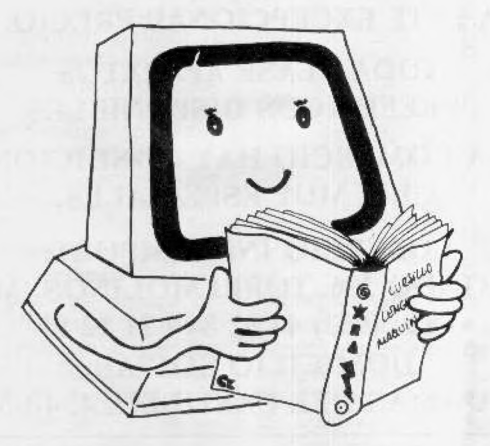
Direccionamiento Inmediato

El operando de la instrucción está situado justo detrás de la propia instrucción, por lo que no requiere acceder a más posiciones de memoria.

En Basic diríamos: LET A= 2 o simplemente A=2.

En C. Máquina decimos: LDA#2 que quiere decir carga en el Acumulador el valor 2. (LoaD Accumulator).

Hemos visto dos cosas al mismo tiempo, el modo de direccionamiento más simple y la instrucción más sencilla: "LDA". Pero si quisiésemos hacer que A tomase el valor de otra posición de memoria que no fuese la siguiente a la instrucción tendríamos que utilizar otro modo de Direccionamiento.



Direccionamiento Absoluto

Se ponen detrás de la instrucción dos bytes que apuntan a la posición de memoria donde se encuentra el dato.

En Basic: A=PEEK (4096) como 4096 pasado a hexadecimal es $1000, en C.M. sería LDA$1000, esta instrucción carga en el acumulador lo que contenga la posición de memoria $1000=4096.

Una advertencia antes de seguir:

Estoy utilizando en los ejemplos BASIC la variable A, pero esto no quiere decir que ésta tenga nada que ver con el registro A de la CPU de nuestro micro, es simplemente por facilitar el equivalente.

Hemos visto la instrucción de carga LDA, pero ahora vamos a conocer una instrucción de almacenamiento STA, que significa "STore Accumulator in memory" y en castellano es "Guarda el acumulador en la memoria..." Con esta instrucción vamos a ver otro ejemplo de direccionamiento absoluto:

En basic: POKE A, 4097 equivale a STA$1001 en máquina.

No necesita explicación porque ya conocéis la función de POKE en basic, y la de STA en máquina es idéntica.

Debemos hacer otra aclaración, en código máquina no escribimos directamente LDA o STA, estos son códigos mnemónicos que facilitan la programación con ayuda de un ensamblador, ya que el lenguaje máquina sólo tiene ceros y unos y sería difícil saber de memoria todos los códigos de las instrucciones si no guardasen algún parecido con la función que realizan.

En C.M. la instrucción LDA#2 sería $A9 $02, y LDA$1000 sería $AD $00 $10 mientras que STA$1001 es $8D $01 $10. Parece bastante más complicado que los mnemónicos ¿verdad?

Para pasar los mnemónicos a C.M. podemos utilizar un ensamblador (existe un cartucho para el VIC y otro para el C-64), o hacer la conversión a mano con ayuda de las tablas que van apareciendo en la revista. Más adelante explicaré cómo se pueden utilizar estas tablas.

Ahora vamos a fijarnos en un detalle, ¿pensabais que me equivoqué al poner $AD $00 $10?... No es así, la CPU lee primero la parte más baja de la dirección, y luego la parte alta, por eso debemos colocarla al revés si lo hacemos sin ayuda del ensamblador, la dirección $AFD9 se escribiría como $D9 y $AF.

Creo que de momento es bastante rollo, hemos visto las instrucciones más sencillas (de carga y almacenamiento) y los dos modos de direccionamiento más fáciles (inmediato y absoluto), si queréis podéis echarle un ojo a cualquier listado de un programa en código máquina y ver estas instrucciones y su aspecto, en el próximo número explicaré los otros modos de direccionamiento y algunas instrucciones más complicadillas...

CUANDO SE TIENE UN COMMODORE 64 ES MUY DIFICIL SER MODESTO

Cuando se tiene un ordenador personal con 64K de memoria, una magnífica resolución, 16 colores, efectos tridimensionales con "sprites", un sonido equivalente al de un sintetizador, un teclado profesional con 62 caracteres gráficos, toda una amplia gama de periféricos profesionales, la más completa serie de programas educativos, profesionales y de video-juegos...; en resumen, cuando se tiene un ordenador personal como no existe ningún otro en el mercado y el más vendido mundialmente, cuando se tiene el Commodore 64, es muy difícil mostrarlo sin que el orgullo se te note.



EL ORDENADOR PERSONAL DE LA FAMILIA MAS POTENTE

– Sistemas de gestión profesionales series 8000 Y 700. – Ordenador portátil SX 64.
– Ordenador personal COMMODORE 64. – Ordenador familiar VIC 20.



MICROELECTRONICA Y CONTROL, S.A.
c/ Taquígrafo Serra, 7, 5.° BARCELONA-29 c/ Princesa, 47, 3.° G MADRID-8

SLOGAN